

Luidger Röckrath

**Assembler-
tricks
auf dem
ColourGenie**

RÖCKRATH
MIKROCOMPUTER

Luidger Röckrath:

Assemblertricks auf dem Colour-Genie

ISBN 3-925074-08-2

Best.-Nr.: 06-008

1. Auflage, Februar 1985

(c) 1985 by

RÖCKRATH
MICROCOMPUTER

Noppiusstr. 19
5100 Aachen

Alle Rechte vorbehalten. Kein Teil dieses Buches darf in irgendeinem Verfahren reproduziert oder in EDV-Anlagen verarbeitet werden.

Herstellung: Fotodruck J. Mainz GmbH, Aachen

Hinweis:

Alle in diesem Buch wiedergegebenen Verfahren und Programme werden ohne Rücksicht auf die Patentlage mitgeteilt und dürfen nicht gewerblich genutzt werden.

Alle Angaben in diesem Buch wurden mit größtmöglicher Sorgfalt erarbeitet bzw. zusammengestellt. Trotzdem sind Fehler nicht ganz auszuschließen. Der Verlag und der Autor sehen sich deshalb gezwungen, darauf hinzuweisen, daß sie weder eine Garantie noch die juristische Verantwortung oder irgendeine Haftung für Folgen, die auf fehlerhafte Angaben zurückgehen, übernehmen können. Für die Mitteilung eventueller Fehler sind Autor und Verlag jederzeit dankbar.

Vorwort

=====

Als Fortsetzung zum Maschinensprachebuch enthält dieses Buch speziell für das Colour-Genie alle wichtigen Maschinenroutinen zur Programmierung der Grafik, der Tonausgabe und der Diskettenstation. Dadurch wird es dem Maschinenprogrammierer zum ersten Mal möglich alle Eigenschaften des Colour-Genies in eigenen Programmen voll auszunutzen.

Alle Routinen werden mit jeweils einem Beispiel erläutert. In jedem Abschnitt folgen größere Beispielprogramme (insgesamt 10), die im Zusammenhang die sinnvolle Anwendung der beschriebenen ROM-Routinen zeigen.

Außerdem enthält dieses Buch die Quellprogramme von zahlreichen Utilities, die als RELOSOFT seit Jahren verkauft werden. An diesen Programmen werden die Möglichkeiten des Eingriffes in Betriebssystem und Basicinterpreter genau erläutert.

Aachen, im Februar 1985

Luidger Röckrath

Inhaltsverzeichnis
=====

| | |
|--|----------------|
| Teil 1: ROM-Routinen und ihre Anwendung | 5 |
| 1.1 : Die Bildschirmausgabe | 6 |
| 1.1.1 Der Textmodus | 6 |
| 1.1.2 Die hochauflösende Grafik | 11 |
| 1.1.3 Der Video-Controller | 25 |
| 1.2 : Die Tonausgabe | 34 |
| 1.2.1 Die Programmierung der Parallelschnittst. | 37 |
| 1.3 : Der Betrieb mit der Diskettenstation | 39 |
| 1.3.1 Allgemeines | 39 |
| 1.3.2 Files eröffnen | 41 |
| 1.3.3 Schreiben und Lesen | 43 |
| 1.3.4 Positionierung des NEXT | 47 |
| 1.3.5 Files Laden, Starten, Löschen | 50 |
| 1.3.6 Files schließen | 52 |
| 1.3.7 Beispiele | 53 |
| 1.3.8 Datum und Uhrzeit | 61 |
| Teil 2: Programmbeschreibungen und Quellprogramme | 64 |
| 2.1.1 Bildschirmeditor SEDIT | 65 |
| 2.1.2 Quellprogramm SEDIT | 72 |
| 2.2.1 Basicerweiterung COLBAS | 82 |
| 2.2.2 Quellprogramm COLBAS | 91 |
| 2.3.1 Feldaufzeichnung ARRAY | 119 |
| 2.3.2 Quellprogramm ARRAY (Kassette) | 121 |
| 2.3.3 Quellprogramm ARRAY (Diskette) | 126 |
| 2.4.1 Kassettenuniversalkopierer COPBIT | 130 |
| 2.4.2 Quellprogramm COPBIT | 138 |
| 2.5.1 Tape --> Disk Converter SAVETAPE | 154 |
| 2.5.2 Originaltexte SAVETAPE | 162 |
| 2.5.3 Quellprogramm SAVETAPE | 166 |
| Literaturverzeichnis | 185 |

Teil 1: ROM-Routinen und ihre Anwendung

=====

```

*****
*
* NAME      : Beschreibung
*
*      E : Einsprung
*      R : Registerbeeinflussung
*      I : Aufrufparameter
*      O : Ergebnisparameter
*
*****

```

Alle folgenden Beschreibungen der ROM-Routinen fangen mit einem derartigen Block an:

NAME : Der Name der Routine, der aus max. 6 Buchstaben und Ziffern besteht. Dahinter kurz zusammengefasst eine Beschreibung der Funktion.

E : Die Einsprungsadresse (ADDRESS)
Die Routine kann entweder mit

```

NAME      EQU      ADDRESS
          CALL      NAME

```

oder

```

          CALL      ADDRESS

```

aufgerufen werden.

R : Registerbeeinflussung.

Hier sind alle Register verzeichnet, die durch die Routine verändert werden können. Nimmt ein spezielles Register immer einen speziellen Wert an, wenn die Routine durchlaufen wird, ist dieser angegeben.

I : Aufrufparameter

Hier ist aufgelistet, in welchen Registern bzw. Speicherzellen, welche Parameter für die Routine übergeben werden müssen.

O : Ergebnisparameter

In den angegebenen Registern und Speicherzellen werden die Ergebnisse der Routine dem Aufrufer zurückgegeben.

Kapitel 1 : Die Bildschirmausgabe

=====

1.1.1 Die Textmodus

Im Textmodus können auf dem Colour-Genie 1000 Zeichen in 25 Zeilen zu je 40 Zeichen dargestellt werden. Die Zeichen können sowohl dem festprogrammierten Zeichengenerator entnommen werden, als auch z. T. frei programmiert werden.

Der Bildwiederholtspeicher des Colour-Genies im Textmodus belegt den Speicherbereich von 4400H-47E7H. Die erste Zeile belegt den Bereich von 4400H-4427H, die zweite den Bereich von 4428H-444FH, usw. In einer Zeile ist das ganz links dargestellte Zeichen der niedrigsten Adresse zuzuordnen.

In jede Speicherzelle kann ein Wert zwischen 0 und 255 (0 - FFH) geschrieben werden. Durch die Videoschaltung wird auf der zugehörigen Bildposition das entsprechende Zeichen dargestellt. Den Werten 0 - 127 entsprechen immer ganz bestimmte Zeichen. Den Werten von 128 - 255 können entweder festprogrammierte Grafikzeichen als auch freiprogrammierte Zeichen zugeordnet werden. Diese Zuordnung wird durch den CHAR-Befehl festgelegt:

| Code | CHAR 0 | CHAR 1 | CHAR 2 | CHAR 3 |
|-------------------|---------------------------------|----------|----------|----------|
| 0-127 (00H-7FH) | festprogrammierte ASCII-Zeichen | | | |
| 128-192 (80H-BFH) | frei PR1 | frei PR1 | fest GR1 | fest GR1 |
| 193-255 (C0H-FFH) | frei PR2 | fest GR2 | frei PR2 | fest GR2 |

Die freiprogrammierbaren Zeichen der Gruppe 1 liegen im Bereich von F400H-F5FFH, die Gruppe 2 im Bereich von F600H-F7FFH. Jedes freiprogrammierbare Zeichen belegt 8 Bytes im Speicher zur Definition des Zeichens. Jedes Byte vereinigt die Bildpunkte von einer Zeile. Als erstes wird die oberste Bildzeile abgespeichert.

```

F400H : Code 80H 1. Zeile
F401H : Code 80H 2. Zeile
F402H : Code 80H 3. Zeile
F403H : Code 80H 4. Zeile
F404H : Code 80H 5. Zeile
F405H : Code 80H 6. Zeile
F406H : Code 80H 7. Zeile
F407H : Code 80H 8. Zeile
F408H : Code 81H 1. Zeile
F409H : Code 81H 2. Zeile
F40AH : Code 81H 3. Zeile
F40BH : Code 81H 4. Zeile
F40CH : Code 81H 5. Zeile
F40DH : Code 81H 6. Zeile
F40EH : Code 81H 7. Zeile
F40FH : Code 81H 8. Zeile
...
F600H : Code C0H 1. Zeile
...
```

F7FFH : Code FFH 8. Zeile

Beispiel:

F400H : 81 42 24 18 18 24 42 81H

Jetzt ergibt ? CHR\$(128) ein großes X.

Die Farben der einzelnen Zeichen sind durch den Farbspeicher definiert, der bei F000H beginnt. Für jedes Zeichen auf dem Bildschirm ist ein Byte reserviert. Von diesen 1024 Bytes sind allerdings nur die unteren 4 Bits verdrahtet, da nur 16 Farben möglich sind. In den Farbspeicher müssen folgende Werte eingeschrieben werden um eine bestimmte Farbe zu erzielen:

| Farbe | Farbcode im Farbspeicher | Argument für COLOUR |
|-------------|-----------------------------|------------------------|
| Grau | 00H = 0 | 10 |
| Cyan | 01H = 1 | 7 |
| Rot | 02H = 2 | 3 |
| Weiß | 03H = 3 | 1 |
| Gelb | 04H = 4 | 4 |
| Grün | 05H = 5 | 2 |
| Orange | 06H = 6 | 5 |
| Hellgelb | 07H = 7 | 11 |
| Blau | 08H = 8 | 6 |
| Hellblau | 09H = 9 | 9 |
| Rotviolett | 0AH = 10 | 15 |
| Blauviolett | 0BH = 11 | 12 |
| Hellgrau | 0CH = 12 | 13 |
| Türkis | 0DH = 13 | 14 |
| Magenta | 0EH = 14 | 8 |
| Hellweiß | 0FH = 15 | 16 |

Die Argumente der COLOUR-Anweisung geben nicht direkt den Farbcode für den Farbspeicher an, sondern werden über eine Tabelle zunächst umcodiert. Diese Tabelle beginnt auf der Adresse 4390H und ist folgendermaßen aufgebaut:

4390H : 03H : COLOUR 1 = Weiß
 4391H : 05H : COLOUR 2 = Grün
 4392H : 02H : COLOUR 3 = Rot
 4393H : 04H : COLOUR 4 = Gelb
 4394H : 06H : COLOUR 5 = Orange
 4395H : 08H : COLOUR 6 = Blau
 4396H : 01H : COLOUR 7 = Cyan
 4397H : 0EH : COLOUR 8 = Magenta
 4398H : 09H : COLOUR 9 = Hellblau
 4399H : 10H : COLOUR 10 = Grau
 439AH : 07H : COLOUR 11 = Hellgelb
 439BH : 0BH : COLOUR 12 = Blauviolett
 439CH : 0CH : COLOUR 13 = Hellgrau
 439DH : 0DH : COLOUR 14 = Türkis
 439EH : 0AH : COLOUR 15 = Rotviolett
 439FH : 0FH : COLOUR 16 = Hellweiß

Folgendes Programmbeispiel zeigt eine sinnvolle Anwendung der freiprogrammierbaren Zeichen. Es ermöglicht die Eingabe der Umlaute und des scharfen s mittels <CTRL> und A, O, U, a, o, u, S und s. Die CTRL-Taste muß losgelassen werden, bevor die Buchstabentaste wird. Bei vielen Colour-Genies sind im Zeichengenerator auf den Codes 1-3 und 5-8 die deutschen Sonderzeichen enthalten. Das Programm SEDIT spricht unter diesen Codes die Umlaute an (s. 2.1)

```

100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* UMLAUTEINGABE/AUSGABE
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200 ;BASIC-ZEIGER:
210 ;
220 KEYVEC EQU 4016H ;KEYBOARD-TREIBER-VEKTOR
230 ;
240 ;ORG 0BF00H ; STARTEN MIT /TM NICHT MOEGL.
250 ;
260 ; INITIALISIERUNG:
270 ;
280 INIT LD HL,(KEYVEC) ; ALTEN KEYBOARDVEKTOR UMSP.
290 LD (NEWKEY+1),HL
300 LD HL,NEWKEY ; NEUEN DEFINIEREN
310 LD (KEYVEC),HL
320 JP 1A19H
330 ;
340 ;
350 ; HAUPTPROGRAMM
360 NEWKEY CALL 0 ; ALTE KEYBOARDROUTINE AUFRUFEN
370 ;
380 AND A ; KEIN ZEICHEN GEDRUECKT
390 RET Z ; JA, -->
400 CP '0'
410 JR Z,CTRL
420 CP 'A' ; < 'A'
430 RET C ; JA, -->
440 CP 'Z'+1 ; <= 'Z'
450 JR C,LETTER ; JA, -->
460 CP 60H ; < SHIFT $
470 RET C ; JA, -->
480 CP 'z'+1 ; <= KLEIN 'Z'
490 JR C,LETTER ; JA, -->
500 RET
510 ;
520 LETTER LD HL,SL ; ZEICHEN MIT SHIFT LOCK
530 XOR (HL) ; VERKNUEPFEN
540 CTRL LD HL,4018H ; CTRL
550 BIT 7,(HL)
560 RES 7,(HL) ; CTRL-BIT LOESCHEN
570 RET Z ; NEIN, -->
580 CP '0' ; '0'
590 JR NZ,CONVER ; NEIN, UMCONVERTIERUNG

```



```

600 ;
610      LD A,(SL)      ; SHIFT LOCK INVERTIEREN
620      XOR 20H
630      LD (SL),A
640      XOR A          ; ZEICHEN LOESCHEN
650      RET
660 ;
670 CONVER LD HL,CTAB1  ; ZEIGER AUF URCODES
680      LD BC,TABLEN   ; TABELLELAENGE
690      PUSH BC        ; RETTEN
700      CPIR          ; IN TABELLE SUCHEN
710      POP BC        ; ZURUECK
720      RET NZ        ; NICHT GEFUNDEN -->
730      ADD HL,BC     ; ADRESSE IN ZIELTAB
740      DEC HL        ; ERRECHNEN
750      LD A,(HL)     ; ZIELZEICHEN LADEN
760      RET
770 ;
780      ;
790 CTAB1  DEFM 'AOUSaous'
800 CTAB3  DEFB 80H     ; CODES FUER AE
810      DEFB 81H     ;      OE
820      DEFB 82H     ;      UE
830      DEFB 86H     ;      SS
840      DEFB 83H     ;      KLEIN AE
850      DEFB 84H     ;      KLEIN OE
860      DEFB 85H     ;      KLEIN UE
870      DEFB 86H     ;      KLEIN SS
880 ;
890 TABLEN EQU 8        ; CTAB2-CTAB1
900 ;
910 SL     DEFB 20H
920 ;
930 ; DEFINITION DER ZEICHEN
940 ;
950      ORG 0F400H
960      DEFB 44H     ; AE
970      DEFB 10H
980      DEFB 28H
990      DEFB 44H
1000     DEFB 7CH
1010     DEFB 44H
1020     DEFB 44H
1030     DEFB 00H
1040     DEFB 44H     ; OE
1050     DEFB 38H
1060     DEFB 44H
1070     DEFB 44H
1080     DEFB 44H
1090     DEFB 44H
1100     DEFB 38H
1110     DEFB 00H
1120     DEFB 44H     ; UE
1130     DEFB 00H
1140     DEFB 44H
1150     DEFB 44H
1160     DEFB 44H
1170     DEFB 44H
1180     DEFB 38H
1190     DEFB 00H

```

| | | |
|------|----------|------|
| 1200 | DEFB 28H | ; AE |
| 1210 | DEFB 00H | |
| 1220 | DEFB 38H | |
| 1230 | DEFB 04H | |
| 1240 | DEFB 3CH | |
| 1250 | DEFB 44H | |
| 1260 | DEFB 3AH | |
| 1270 | DEFB 00H | |
| 1280 | DEFB 00H | ; OE |
| 1290 | DEFB 28H | |
| 1300 | DEFB 00H | |
| 1310 | DEFB 38H | |
| 1320 | DEFB 44H | |
| 1330 | DEFB 44H | |
| 1340 | DEFB 38H | |
| 1350 | DEFB 00H | |
| 1360 | DEFB 00H | ; UE |
| 1370 | DEFB 28H | |
| 1380 | DEFB 00H | |
| 1390 | DEFB 44H | |
| 1400 | DEFB 44H | |
| 1410 | DEFB 4CH | |
| 1420 | DEFB 34H | |
| 1430 | DEFB 00H | |
| 1440 | DEFB 38H | ; SS |
| 1450 | DEFB 44H | |
| 1460 | DEFB 44H | |
| 1470 | DEFB 58H | |
| 1480 | DEFB 44H | |
| 1490 | DEFB 44H | |
| 1500 | DEFB 58H | |
| 1510 | DEFB 40H | |
| 1520 | END INIT | |

| Farbe | b n+1 | b n |
|---------|-------|-----|
| schwarz | 0 | 0 |
| blau | 0 | 1 |
| rot | 1 | 0 |
| grün | 1 | 1 |

Die 2 Bits von je 4 Bildpunkten sind zu einem Byte zusammengefaßt, und zwar jeweils 4 in x-Richtung aufeinanderfolgende Bildpunkte. Der Punkt links belegt die beiden höchstwertigen Bits, der Bildpunkt rechts die beiden niederwertigen. Die Punkte einer Zeile sind von links nach rechts in aufeinanderfolgenden Speicherzellen abgelegt. Jede Bildzeile belegt 40 (=28H) Bytes ($160 / 4 = 40$). Die 102 Bildzeile sind hintereinander im Speicher abgelegt, beginnend mit der Zeile mit der y-Koordinate 0:

| 0,0 ! 1,0 ! 2,0 ! 3,0 | 4,0 ! 5,0 ! 6,0 ! 7,0 ! | 156,0 !157,0 !158,0 !159,0 |
|---|---|---|
| 4800: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | 4801: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | ..4827: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! |
| 0,1 ! 1,1 ! 2,1 ! 3,1 | 4,1 ! 5,1 ! 6,1 ! 7,1 | 156,1 !157,1 !158,1 !159,1 |
| 4828: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | 4829: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | ..484F: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! |
| 0,2 ! 1,2 ! 2,2 ! 3,2 | 4,2 ! 5,2 ! 6,2 ! 7,2 ! | 156,2 !157,2 !158,2 !159,2 |
| 4850: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | 4851: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | ..4877: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! |
| 0,99 ! 1,99 ! 2,99 ! 3,99 | 4,99 ! 5,99 ! 6,99 ! 7,99 ! | 156,99 !157,99 !158,99 !159,99 |
| 5778: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | 5779: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | ..579F: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! |
| 0,100! 1,100! 2,100! 3,100 | 4,100! 5,100! 6,100! 7,100! | 156,100!157,100!158,100!159,100 |
| 57A0: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | 57A1: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | ..57C7: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! |
| 0,101! 1,101! 2,101! 3,101 | 4,101! 5,101! 6,101! 7,101! | 156,101!157,101!158,101!159,101 |
| 57C8: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | 57C9: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! | ..57EF: !b7 !b6 !b5 !b4 !b3 !b2 !b1 !b0 ! |

1.1.2 Bildschirmausgabe: Die hochauflösende Grafik Seite 12

Viele ROM-Routinen können zur Erstellung von Grafiken, genutzt werden, die in ihrer Funktion, den Basicanweisungen gleichen Namens entsprechen:

```
*****
*
* FGR      : hochauflösende Grafik einschalten
*
*      E : 38A9H
*      R : AF, BC
*      I : /
*      O : /
*
*****
```

Funktion:

Die hochauflösende Grafik wird eingeschaltet, dazu wird das Bit 5 von Port FFH gesetzt und der CRTC speziell initialisiert.

Beispiel:

```
CALL    FGR
```

```
*****
*
* LGR      : hochauflösende Grafik ausschalten
*
*      E : 38B0H
*      R : AF, BC
*      I : /
*      O : /
*
*****
```

Funktion:

Die hochauflösende Grafik wird ausgeschaltet, dazu wird das Bit 5 von Port FFH zurückgesetzt und der CRTC speziell initialisiert.

Beispiel:

```
CALL    LGR
```

```
*****
*
* BGRD   : Hintergrund einschalten
*
*       E : 38B7H
*       R : AF, B
*       I : /
*       O : /
*
*****
```

Funktion:

Der Hintergrund (schwarz durch Magenta ersetzt) wird eingeschaltet, dazu wird das Bit 2 von Port FFH gesetzt.

Beispiel:

```
CALL    FGR
```

```
*****
*
* NBGRD  : Hintergrund ausschalten
*
*       E : 38BBH
*       R : AF, B
*       I : /
*       O : /
*
*****
```

Funktion:

Der Hintergrund (schwarz durch Magenta ersetzt) wird ausgeschaltet, dazu wird das Bit 2 von Port FFH zurückgesetzt.

Beispiel:

```
CALL    NBGRD
```

```
*****
*
* FCLS   : Bildschirm mit Farbe füllen
*
*       E : 384DH
*       R : AF, BC, DE
*       I : A = Farbcode (0 = sch, 1 = bl, 2 = rot, 3 = gr)*
*       O : /
*
*****
```

Funktion:

Der gesamte HGR-Bildschirm wird mit der angegebenen Farbe gefüllt. Dazu wird der Speicherbereich von 4800 bis 57FF (!) mit 00H (schwarz), 55H (blau), AAH (rot) oder FFH (grün) gefüllt.

Beispiel:

```
LD      A,3
CALL    FCLS                ; Bildschirm grün
```

```
*****
*
* FCOLOU : Schreibfarbe festlegen
*
*      E : 38DAH
*      R : AF
*      I : A = Farbcode (0 = sch, 1 = bl, 2 = rot, 3 = gr)*
*      O : /
*
*****
```

Funktion:

Mit diesem Unterprogramm wird die Schreibfarbe für alle folgenden Operationen in der Grafik festgelegt. Nach Überprüfung auf Bereichsüberschreitung, die einen FC-Fehler hervorruft, wird die Schreibfarbe in (4313H) abgelegt.

Beispiel:

```
LD      A,1                ; Farbcode für blau
CALL    FCOLOU             ; Schreibfarbe festlegen
```

```
*****
*
* CIRCLE : Kreis zeichnen
*
*      E : 3B09H
*      R : AF, BC, DE
*      I : B = x-Koordinate des Mittelpunktes
*           D = y-Koordinate des Mittelpunktes
*           A = Radius
*      O : /
*
*****
```

Funktion:

entspricht Basic: CIRCLE B,D,A.

Beispiel:

```
LD      B,80                ; X = 80
LD      D,50                ; Y = 50
LD      A,30                ; R = 30
CALL    CIRCLE              ; CIRCLE 80,50,30
```

```

*****
*
* PAINT : Fläche mit Farbe füllen
*
*      E : 3E78H
*      R : AF, BC, DE
*      I : E = x-Koordinate
*          D = y-Koordinate
*          (431DH) = Anzahl der Begrenzungsfarben + 1
*          (4319H) = Füllfarbe
*          (431AH) = Begrenzungsfarbe 1
*          (431BH) = Begrenzungsfarbe 2
*      O : /
*
*****

```

Funktion:

entspricht BASIC: PAINT E,D,(4319H)+1,(431AH)+1,(431BH)+1
 Es können keine, eine oder zwei Begrenzungsfarben angegeben werden. (431DH) muß dann die Werte 1, 2 oder 3 enthalten.

Beispiel:

```

LD      E,80          ; x-Koordinate
LD      D,50          ; y-Koordinate
LD      A,2           ; 1 Begrenzungsfarbe
LD      (431DH),A
LD      A,1           ; Füllfarbe = blau
LD      (4319H),A
LD      A,2           ; Begrenzungsfarbe = rot
LD      (431AH),A
CALL    PAINT         ; PAINT D,E,2,3

```

```

*****
*
* DETOHL : Line von (D,E) nach (H,L) ziehen
*
*      E : 3C1FH
*      R : AF, BC, DE, HL, BC', DE', HL'
*      I : D = x-Koordinate 1
*          E = y-Koordinate 1
*          H = x-Koordinate 2
*          L = y-Koordinate 2
*      O : /
*
*****

```

Funktion:

Von dem Punkt mit den Koordinaten (D,E) wird eine Linie zu dem Punkt mit den Koordinaten (H,L) gezogen. Die Farbe muß zuvor mit FCOLOU definiert werden.

Beispiel:

```

LD      D,1          ; x1 = 1
LD      E,1          ; y1 = 1
LD      H,158        ; x2 = 158

```

```
LD      L,100          ; y2 = 100
CALL    DETOHL         ; Linie (D,E) nach (H,L)
```

```
*****
*
* PLOTHL : Punkt setzen
*
*      E : 3B8AH
*      R : AF, BC, DE, HL
*      I : L = x-Koordinate
*          H = y-Koordinate
*      O : /
*
*****
```

Funktion:

Der Punkt mit den Koordinaten (D,E) wird mit der Farbe, die zuvor durch FCOLOU definiert wurde, gesetzt. Durch Einsprung bei PLOTHL + 3 kann der Farbcode in A übergeben werden.

Beispiel:

```
LD      L,80           ; x = 80
LD      H,50           ; y = 50
CALL    PLOTHL         ; Punkt (L,H) setzen
```

```
*****
*
* CPOINT : Punkt abfragen
*
*      E : 3F3AH
*      R : AF, BC, DE, HL
*      I : L = x-Koordinate
*          H = y-Koordinate
*      O : A = Farbcode des Punktes
*
*****
```

Funktion:

Die Farbe des Punktes mit den Koordinaten (L,H) wird abgefragt und im Akku zurückgegeben.

Beispiel:

```
LD      L,80           ; x = 80
LD      H,50           ; y = 50
CALL    CPOINT         ; Punkt abfragen
CP      1              ; Farbe blau ?
JR      Z,BLAU         ; ja, -->
```


Die ersten beiden Beispiele zeigen wie Bilder in der hochauflösenden Grafik von Maschinenprogrammen aufgebaut werden können. Beispiel 1 malt konzentrische Kreise mit wechselnden Farben und demonstriert insbesondere die Anwendung der Unterprogramme CIRCLE und PAINT.

```

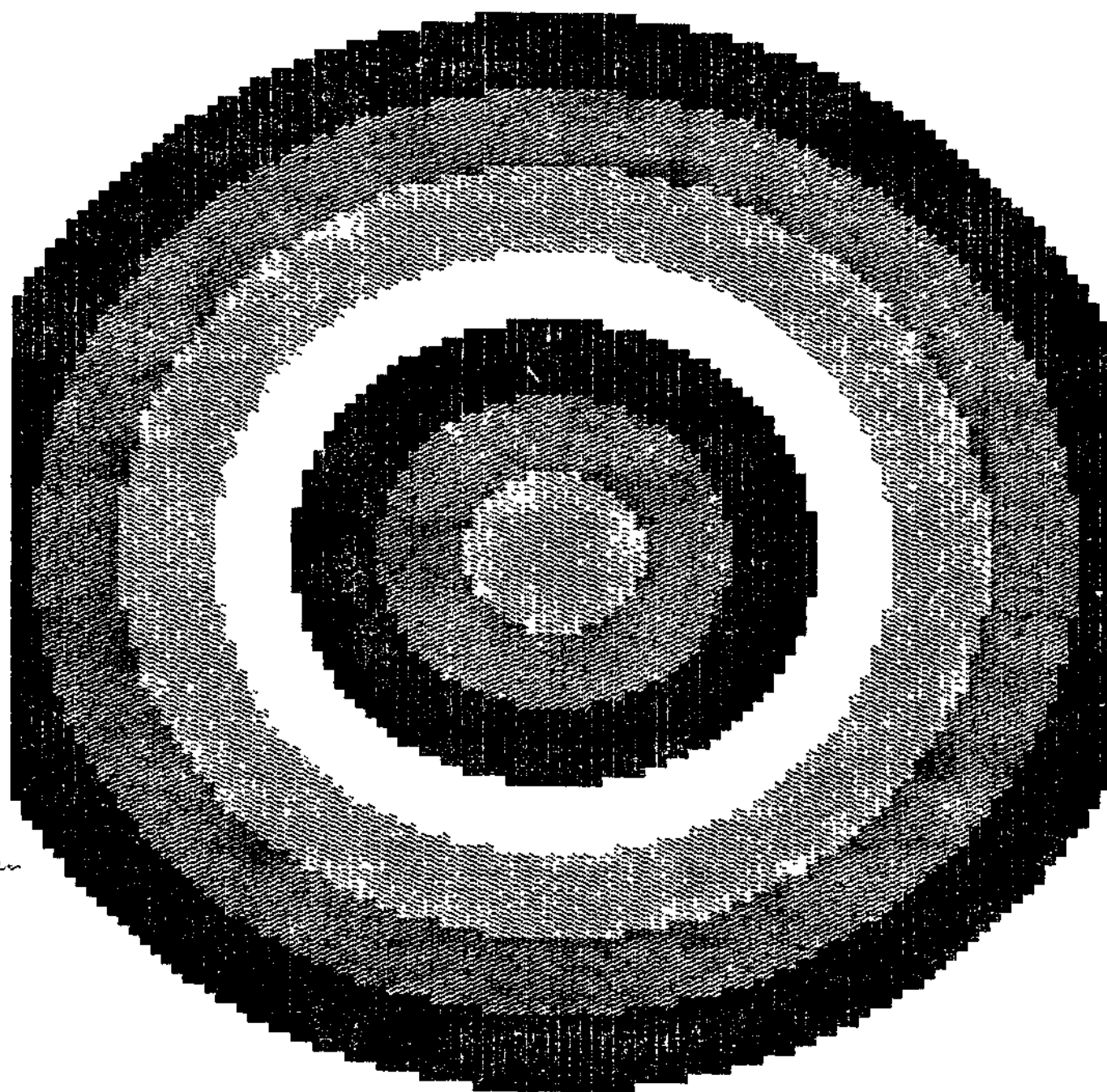
100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* KONZENTRISCHE KREIS IN FGR
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200 FCOLOU EQU 38DAH ; SCHREIBFARBE WAEHLEN
210 FCLS EQU 384DH ; BILDSCHIRM LOESCHEN
220 FGR EQU 38A9H ; FGR ANWAEHLEN
230 LGR EQU 38B0H ; LGR ANWAEHLEN
240 BGRD EQU 38B7H ; BACKGROUND EIN
250 NBGRD EQU 38BBH ; BACKGROUND AUS
260 CIRCLE EQU 3B09H ; KREIS ZEICHNEN
270 PAINT EQU 3E78H ; FLAECHEN AUSFUELLEN
280 ;
290 ANZPNT EQU 431DH ; ANZAHL DER ARG. FUER PAINT
300 FUELLF EQU 4319H ; FUELLFARBE
310 BEGRF1 EQU 431AH ; BEGRENZUNGSFARBE
320 BEGRF2 EQU 431BH ; BEGRENZUNGSFARBE
330 ;
340 ; PROGRAMMSTART
350 ;
360 ORG 7000H
370 START CALL FGR ; GRAFIK EIN
380 LD A,0 ; FCLS
390 CALL FCLS
400 LD A,1 ; FCOLOUR 1
410 CALL FCOLOU
420 LD C,8 ; RADIUS
430 LOOP LD A,C ; FCOLOUR RADIUS / 8
440 RRCA
450 RRCA
460 RRCA
470 AND 3
480 JR NZ,L1 ; KEIN SCHWARZ, -->
490 INC A ; SCHWARZ --> BLAU
500 L1 CALL FCOLOU
510 LD B,80 ; X = 80
520 LD D,51 ; Y = 51
530 LD A,C ; RADIUS -> A
540 PUSH BC ; RADIUS RETTEN
550 AND A
560 CALL CIRCLE ; CIRCLE 80,51,B
570 POP BC
580 LD D,51 ; Y=51
590 LD A,C ; X=80+RADIUS-4
600 ADD A,80-4
610 LD E,A
620 LD A,3 ; ANZAHL DER ARG FUER PAINT
630 LD (ANZPNT),A

```

```

640      LD      A,C      ; FARBE = RADIUS / 8
650      RRCA
660      RRCA
670      RRCA
680      AND     3
690      LD      (FUELLF),A ; FUELLFARBE
700      LD      (BEGRF1),A ; BEGRENZUNGSFARBE 1
710      DEC     A      ; LETZTE KREISFARBE
720      AND     3
730      JR      NZ,L2   ; KEIN SCHWARZ -->
740      INC     A      ; SCHWARZ --> BLAU
750 L2     LD      (BEGRF2),A ; BEGRENZUNGSFARBE 2
760      PUSH    BC
770      CALL    PAINT   ; PAINT E,D,C,C,C-1
780      POP     BC
790      LD      A,C      ; RADIUS = RADIUS + 8
800      ADD     A,8
810      LD      C,A
820      CP      64      ; RADIUS = 64
830      JR      NZ,LOOP ; NEIN -->
840      CALL    49H     ; WARTETASTENDRUCK
850      RET
860      END      START

```



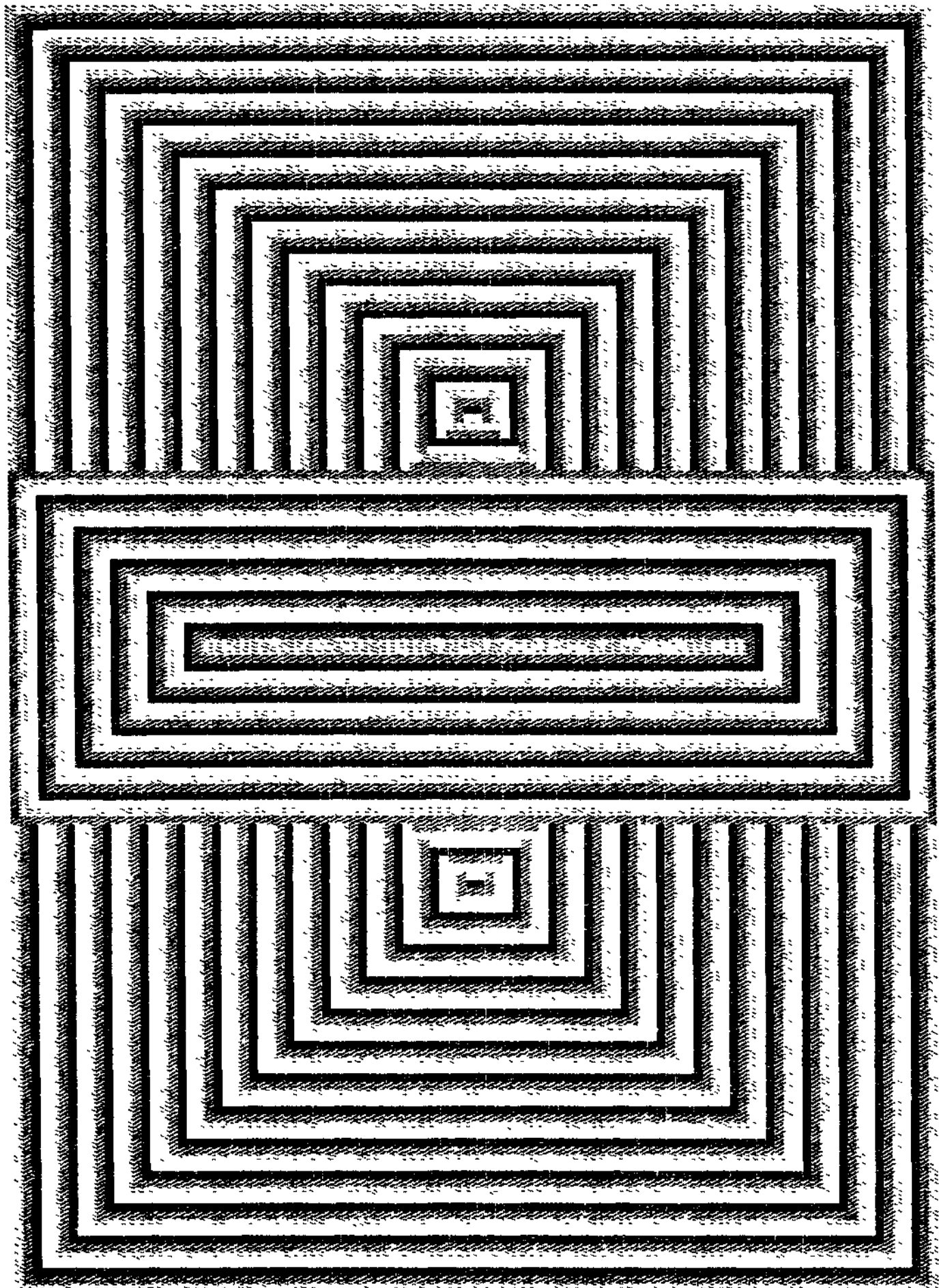
Beispiel 2 füllt den Bildschirm mit ineinander verschachtelten Rechtecken. Die Unregelmäßigkeiten dabei ergeben sich aus dem Wachsen des Parameters B über 51. Durch Drücken der RETURN-Taste nach Fertigstellung des ersten Bildes (wird erst nach einiger Zeit akzeptiert), wird der Background abgeschaltet. Nach nochmaligem Drücken der RETURN-Taste werden die Farben verändert. Durch ein weiteres Drücken der RETURN-Taste kann danach wieder der Background eingeschaltet werden. Schließlich kann durch Drücken der RETURN-Taste das Programm beendet werden. Das Beispiel zeigt insbesondere die Anwendung der Unterprogramme DETOHL, PLOTHL und CPOINT. Dabei ist besonders zu beachten, daß DETOHL die X-Endkoordinate in H und die Y-Endkoordinate in L erwartet, während PLOTHL und CPOINT die X-Koordinate in L und die Y-Koordinate in H erwarten.

```

100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* RECHTECKE IN FGR AUSGEBEN
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200 FCOLOU EQU 38DAH
210 FCLS EQU 384DH
220 FGR EQU 38A9H
230 LGR EQU 38B0H
240 BGRD EQU 38B7H
250 NBGRD EQU 38BBH
260 DETOHL EQU 3C1FH ; LINIE VON (D,E) NACH (H,L)
270 PLOTHL EQU 3B8AH ; PUNKT AUF (L,H) SETZEN
280 CPOINT EQU 3F3AH ; PUNKT (L,H) ABFRAGEN
290 ;
300 ; PROGRAMMSTART
310 ;
320 ORG 7000H
330 START CALL FGR ; GRAFIK EIN
340 LD A,0 ; FCLS
350 CALL FCLS
360 CALL BGRD ; BGRD
370 LD B,1 ; ZAEHLER
380 LOOP LD A,B
390 AND 3
400 CALL FCOLOU ; FCOLOUR B
410 LD D,B ; X1=B
420 LD E,B ; Y1=B
430 LD A,159 ; X2=159-B
440 SUB B
450 LD H,A
460 LD L,B ; Y2=B
470 PUSH BC
480 PUSH HL ; ENDKOOR.
490 CALL DETOHL ; (D,E) --> (H,L)
500 POP DE ; JETZT ANFKOOR.
510 POP BC
520 LD A,101 ; Y2=101-B

```

```
530      SUB      B
540      LD       L,A
550      LD       H,D      ; X2=X1
560      PUSH     BC
570      PUSH     HL      ; ENDKOOR.
580      CALL     DETOHL   ; (D,E) --> (H,L)
590      POP      DE      ; JETZT ANFKOOR.
600      POP      BC
610      LD       L,E      ; Y2=Y1
620      LD       H,B      ; X2=B
630      PUSH     BC
640      PUSH     HL      ; ENDKOOR.
650      CALL     DETOHL   ; (D,E) --> (H,L)
660      POP      DE      ; JETZT ANFKOOR.
670      POP      BC
680      LD       L,B      ; Y2=B
690      LD       H,B      ; X2=B
700      PUSH     BC
710      CALL     DETOHL   ; (D,E) --> (H,L)
720      POP      BC
730      INC      B      ; GROESSE - 1
740      JR       NZ,LOOP ; GROESSE > 0 -->
750      CALL     49H      ; WARTEN TASTENDRUCK
760      CALL     NBGRD     ; BACKGROUND AUS
770      CALL     49H      ; WARTEN TASTENDRUCK
780      ;
790      ; FARBEN ROTIEREN:
800      ; SCHWARZ --> BLAU --> ROT --> GRUEN -->
810      ;
820      LD       HL,0      ; H = Y, L = X, X=0, Y=0
830 LOOP2  PUSH     HL
840      CALL     CPOINT   ; FARBE BEST.
850      POP      HL
860      INC      A      ; FARBE + 1
870      AND      3
880      CALL     FCOLOU   ; NEUE FARBE SETZEN
890      PUSH     HL
900      CALL     PLOTHL   ; PUNKT IN NEUER FARBE SETZEN
910      POP      HL
920      INC      L      ; X + 1
930      LD       A,L      ; X = 160
940      CP       160
950      JR       NZ,LOOP2 ; NEIN, -->
960      LD       L,0      ; X = 0
970      INC      H      ; Y + 1
980      LD       A,H      ; Y = 102
990      CP       102
1000     JR       NZ,LOOP2 ; NEIN, -->
1010     CALL     49H      ; WARTEN TASTENDRUCK
1020     CALL     BGRD     ; BACKGROUND EIN
1030     CALL     49H      ; WARTEN TASTENDRUCK
1040     RET
1050     END      START
```



1.1.2 Bildschirmausgabe: Die hochauflösende Grafik Seite 23

Das letzte Beispiel ermöglicht den Ausdruck des Grafikbildschirm auf einem Drucker. Die Farben werden in Graustufen umgesetzt. Die SteuerCodes für den Drucker sind für alle EPSON- und STAR-Drucker geeignet. Bei anderen Druckern müssen nur folgende SteuerCodes geändert werden:

1BH,41H,n = Vorschub auf n-Punkte einstellen
 1BH,4CH,C0H,03H = 960 Bytes in Grafik folgen

```

100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* FGR AUSGABE AUF DRUCKER
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200 PRCHAR EQU 050EH ; ZEICHEN A AUF DRUCKER
210 BREITE EQU 40 ; 40 ZEICHEN PRO ZEILE
220 BYPDOT EQU 4 ; BYTE AUF DRUCKER PRO DOT
230 MASKE EQU 0F0H ; MASKE
240 HGRRAM EQU 4800H ; SPEICHER FUER FGR
250 ESC EQU 27 ; ESC = 27
260 CR EQU 13
270 ;
280 START LD A,ESC ; ESC,A,BYPDOT AUF DRUCKER
290 CALL PRINT ; DEFINIERT ZEILENVORSCHUB
300 LD A,'A' ; IN PUNKTEN
310 CALL PRINT
320 LD A,BYPDOT
330 CALL PRINT
340 LD HL,HGRRAM
350 ZEILE LD B,BREITE ; 40 ZEICHEN / ZEILE
360 LD A,ESC ; ESC,L,C0H,3
370 CALL PRINT ; ES FOLGEN 960 GRAPHIC-BYTES
380 LD A,'L'
390 CALL PRINT
400 LD A,0C0H
410 CALL PRINT
420 LD A,3
430 CALL PRINT
440 BYTE LD A,(HL) ; 1 BYTE AUS FGR
450 INC HL
460 LD C,4 ; 4 DOTS PRO BYTE
470 DOT RLCA ; NAECHSTEN PUNKT
480 RLCA ; NACH BIT 0 UND 1
490 PUSH AF ; BYTE RETTEN
500 AND 3 ; 1 DOT HERAUSBLENDEN
510 ADD A,A ; * 2
520 PUSH AF
530 ADD A,A ; * 4
540 POP DE
550 ADD A,D ; A = FARBCODE * 6
560 PUSH HL
570 LD HL,MUSTER
580 LD E,A
590 LD D,0
600 ADD HL,DE ; HL ZEIGT AUF MUSTER
  
```

```

610      PUSH      BC
620      LD        B,6      ; 6 BYTES DES MUSTERS
630 OUT   LD        A,(HL)   ; MUSTERBYTE LADEN
640      AND        MASKE    ; MIT MASKE UNDIEREN
650      CALL       PRINT    ; AUSGEBEN
660      INC        HL
670      DJNZ       OUT
680      POP        BC      ; ZAEHLER ZURUECK
690      POP        HL      ; ZEIGER AUF FGR-RAM ZURUECK
700      POP        AF      ; BYTE ZURUECK
710      DEC        C      ; 4 PUNKTE / BYTE BEARBEITET
720      JR         NZ,DOT   ; NEIN, -->
730      DJNZ       BYTE    ; ZEILE FERTIG, NEIN -->
740      LD        A,CR      ; VORSCHUB BYPDOT
750      CALL       PRINT    ; PUNKTE
760      LD        DE,57F0H ; ENDE FGR-RAM PRINT
770      RST        18H
780      JR         NZ,ZEILE ; NEIN --> WEITER
790      RET
800      ; DRUCKT ZEICHEN IN A
810 PRINT   PUSH     BC
820      LD        C,A
830      CALL       PRCHAR
840      POP        BC
850      RET
860      ;
870      ; MUSTER ZUR UMSETZUNG DER FARBE
880      ;
890 MUSTER  DEFB     0      ; SCHWARZ
900      DEFB     0
910      DEFB     0
920      DEFB     0
930      DEFB     0
940      DEFB     0
950      DEFB     24H      ; BLAU:      X  X
960      DEFB     48H      ;          X  X
970      DEFB     90H      ;          X  X
980      DEFB     24H      ;          X  X
990      DEFB     48H      ;          X  X
1000     DEFB     90H      ;          X  X
1010     DEFB     6CH      ; ROT      XX XX
1020     DEFB     0B4H     ;          X XX X
1030     DEFB     0D8H     ;          XX XX
1040     DEFB     6CH      ;          XX XX
1050     DEFB     0B4H     ;          X XX X
1060     DEFB     0D8H     ;          XX XX
1070     DEFB     0FCH     ; GRUEN    XXXXXX
1080     DEFB     0FCH     ;          XXXXXX
1090     DEFB     0FCH     ;          XXXXXX
1100     DEFB     0FCH     ;          XXXXXX
1110     DEFB     0FCH     ;          XXXXXX
1120     DEFB     0FCH     ;          XXXXXX
1130     END          START

```


1.1.3 Der Video-Controller (CRTC)

Die Umwandlung des Inhaltes des Bildwiederholtspeichers in ein Videosignal wird beim Colour-Genie von einem hochintegrierten Videocontroller (CRTC) übernommen. Dieser Chip hat 16 Register mit denen er programmiert werden kann. Über den Systembus ist er mit 2 Adressen anzusprechen: 0FAH (= 250) und 0FBH (= 251). In die Adresse 250 muß zunächst die Registernummer geschrieben werden, bevor mit einem OUT-Befehl auf die Adresse 251 das entsprechende Register belegt werden kann.

Die 16 Register des CRTC haben folgende Funktion:

| Reg | Funktion | Adr. | | Initwert | |
|-----|---------------------------------------|------|------|----------|-----|
| | | LGR | FGR | LGR | FGR |
| 0 | Horizontale Abtastgeschwindigkeit | 42FF | 430F | 70 | 70 |
| 1 | Zeichen pro Zeile | 42FE | 430E | 40 | 40 |
| 2 | Horizontale Synchronisationsposition | 42FD | 430D | 52 | 52 |
| 3 | Länge der Zeilensynchronisation | 42FC | 430C | 150 | 150 |
| 4 | Vertikale Gesamtzahl der Zeilen | 42FB | 430B | 38 | 126 |
| 5 | Vertikale Justierung (Bildfang) | 42FA | 430A | 0 | 31 |
| 6 | Zeilenzahl (dargestellte Zeilen) | 42F9 | 4309 | 25 | 102 |
| 7 | Vertikale Synchronisationsposition | 42F8 | 4308 | 31 | 116 |
| 8 | Interlace und Skew | 42F7 | 4307 | 160 | 32 |
| 9 | Zeichen-End-Raster | 42F6 | 4306 | 7 | 1 |
| 10 | Cursor Blinken, Cursor Start Raster | 42F5 | 4305 | 196 | 32 |
| | Bit 0-3 : Cursorstartraster | | | | |
| | Bit 5-6 : = 00 Cursor blinkt nicht | | | | |
| | : = 01 Cursor aus | | | | |
| | : = 10 Cursor blinkt schnell | | | | |
| | : = 11 Cursor blinkt langsam | | | | |
| 11 | Cursor End Raster | 42F4 | 4304 | 7 | 0 |
| 12 | Bildwiederholungspeicheradresse (MSB) | 42F3 | 4303 | 4 | 8 |
| 13 | " (LSB) | 42F2 | 4302 | 0 | 0 |
| 14 | Cursoradresse | 42F1 | 4301 | 0 | 0 |
| 15 | " | 42F0 | 4300 | 1 | 0 |

Bemerkungen:

zu: Horizontale Abtastgeschwindigkeit

Erhöhen des Wertes macht das Bild schmaler, Erniedrigen breiter. Änderungen sind aber nur in einem sehr schmalen Bereich möglich, da sonst die Synchronisation verloren geht.

zu: Horizontale Synchronisationsposition

Durch Verändern dieses Wertes kann das Bild auf dem Bildschirm horizontal verschoben werden. Größerer Wert bewirkt Linksverschieben.

zu: Vertikale Synchronisationsposition

siehe oben. Größerer Wert bewirkt Hochschieben des Bildes.

zu: Zeichenendraster

alleinige Änderung führt sofort zu Desynchronisation

zu: Cursor Blinken und Cursorstartraster / -endraster

Die Werte für diese beiden Register nicht hier, sondern in die Speicherzellen 4019H (Endraster) und 401AH (Startraster und Blinken) eintragen. Die RAM-Tabelle wird nur bei der Initialisierung benutzt. Die Videoausgabe über OUTCH setzt bei jedem Aufruf durch die Routine CURSET (s. u.) beide Register auf die in diesen Speicherzellen befindlichen Werte.

zu: Bildschirmspeicheradresse

Hier muß die Adresse - 4000H abgelegt werden.

zu: Cursoradresse

Die Cursoradresse muß absolut (also nicht abzüglich 4000H, s. o.) hier abgelegt werden. Ändern der Werte in der RAM-Tabelle ist auch hier sinnlos, da die OUTCH-Routine hier bei jedem Aufruf die Cursorposition (4020H) einträgt.

Für die Werte aller 16 Register ist für beide Grafikmodi im RAM eine Tabelle enthalten, die die Werte allerdings in umgekehrter Reihenfolge enthält. Diese beiden Tabellen werden beim Systemstart aus dem ROM geladen (ROM-Tabelle LGR: 3800H, FGR: 3810H) und durch die Anweisungen LGR und FGR werden die CRTC-Register belegt und außerdem wird ein Bit von Port 0FFH umgesetzt. Im ROM sind übrigens noch je eine Tabelle für die NTSC-Norm vorhanden (LGR: 3823H, FGR: 3833H).

Außer durch die Anweisungen LGR und FGR werden die CRTC-Register auch bei jedem Wiedereintritt in die Hauptschleife mit den LGR-Werten neu belegt. Daher können durch POKes in die RAM-Tabelle im Direktmode Änderungen vorgenommen werden, die beim Erscheinen des 'Ready' wirksam werden. Änderungen der Cursorparameter und -position sind hierdurch aber nicht möglich (s. o.).

Will man die RAM-Tabelle per Basicprogramm ändern, muß man zur Neuinitialisierung des CRTC den gerade gewählten Grafikmode noch einmal aktivieren.

Von Maschinenprogrammen aus können mit folgenden Anweisungen die CRTC-Register belegt werden:

```
LD      A,1           ; Register 1 anwählen
OUT     (0FAH),A
LD      A,42          ; Zeichen / Zeile = 42
OUT     (0FBH),A
```

Außerdem können folgende Unterprogramme benutzt werden:

```
*****
*
* CRTC2S : 2 Register des CRTC setzen
*
*      E : 3047H
*      R : AF
*      I : A = Registernummer
*          HL = Wert (H für Reg A, L für Reg A+1
*      O : /
*
*****
```

Funktion:

Zwei Register des CRTC werden neu belegt. Reg A erhält den Wert von H, Reg A+1 den von L.

Beispiel:

```
LD      HL,(4020H)      ; Cursoradresse laden
LD      A,14            ; Reg 14 (Cursoradresse)
CALL    CRTC2S          ; Cursoradresse CRTC setzen
```

```
*****
*
* CURSET : Cursor setzen
*
*      E : 305FH
*      R : AF
*      I : HL = Cursoradresse
*      O : /
*
*****
```

Funktion:

Reg 10 des CRTC wird den Cursorparametern aus (4019H) und (401AH) geladen und Reg 14 mit der Cursoradresse. Außerdem wird auf der Cursorposition im Farbspeicher die augenblickliche Schreibfarbe (4023H) eingesetzt.

Beispiel:

```
LD      HL,4450H        ; Cursor auf 4450H
CALL    CURSET
```

```
*****
*
* CURRES : Cursor löschen
*
*      E : 3059H
*      R : AF
*      I : HL = Cursoradresse
*      O : /
*
*****
```

Funktion:

Reg 10 des CRTC wird mit 20H, Reg 11 mit 7 und Reg 14 mit der Cursoradresse geladen. Ansonsten siehe oben.

Beispiel:

```
LD      HL,4450H      ; Cursor auf 4450H
CALL    CURRES        ; löschen
```

Mit folgendem Beispielprogramm wird das Bildschirmformat im Textmodus auf 30 Zeilen a 50 Zeichen erweitert. Dazu muß der CRTC umprogrammiert werden. Außer den Zeichen pro Zeile und der Anzahl der dargestellten Zeilen muß die Position der horizontalen und vertikalen Synchronisation geändert werden, um das Bild nach links oben zu verschieben. Diese Programmierung erfolgt bei der Initialisierung.

Würde man nur den CRTC derartig umprogrammieren, würde das Bild ziemlich konfus aussehen, denn der Videotreiber des Basicinterpreters muß das Bildformat kennen, um den Bildwiederholtspeicher richtig zu füllen. Also muß ein völlig neuer Videotreiber geschrieben werden, in dem alle Parameter, die aus dem Bildformat hergeleitet sind, geändert sind. Folgendes Beispielprogramm enthält einen solchen Videotreiber. Dazu wurde mit dem NEWDOS Disassembler (DISASSEM) mit der STD-Option aus dem Treiber im ROM (3039H - 3222H, sowie 35F8H - 3652H) ein Quellprogramm erzeugt, das dann an allen erforderlichen Stellen geändert wurde. Schließlich wurden die überflüssigen Programmteile wieder entfernt und stattdessen Sprünge und Aufrufe ans ROM eingesetzt. So wurde der nachfolgend aufgelistete Videotreiber erstellt. Aus dieser Endstehungsweise erklären sich die merkwürdigen Labels (AAA, AAB, ...) und das fast gänzliche Fehlen von Kommentaren. Lediglich einige strukturierende Überschriften wurden eingefügt. Wer an tiefergehenden Kommentaren interessiert ist, kann im ROM-Listing den Originalvideotreiber durcharbeiten.

```

100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* Bildschirm auf 30 Zeilen a 50 Zeichen
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200          ORG      0BC00H
210 DISVEC  EQU      401EH
220 RST18V  EQU      4007H
230 VIDRAM  EQU      4400H
240 CURSET  EQU      305FH
250 CURRET  EQU      3059H
260 COLSET  EQU      307AH
270 CLS     EQU      01C9H
280          ;
290 CPL      EQU      50      ; ZEICHEN PRO ZEILE
300 LINES    EQU      30      ; ZEILENANZAHL
310 GESAMT   EQU      1500    ; = LINES*CPL = GESAMTZAHL DER
                                ; ZEICHEN AUF DEM BILDSCHIRM

330 FCERR    EQU      1E4AH
340 CURSOR   EQU      4020H
350 LAST     EQU      VIDRAM+GESAMT
360 FARBE    EQU      0F000H
370          ;
380 INIT     LD        A,CPL
390          LD        (17150),A      ; CPL IN CRT-TABELLE
400          LD        A,LINES

```

```

410      LD      (17145),A      ; LINES IN CRT-TABELLE
420      LD      A,33
430      LD      (17144),A      ; HORIZONTALE JUSTIERUNG
440      LD      A,56
450      LD      (17149),A      ; VERTIKALE JUSTIERUNG
460      ;
470      LD      HL,VIDEO
480      LD      (DISVEC),HL
490      ;
500      LD      HL,(RST18V)
510      LD      (RST18C+1),HL
520      LD      HL,RST18N
530      LD      (RST18V),HL
540      CALL    CLS
550      JP      1A19H
560      ;
570  ABB      INC      HL
580      PUSH    HL
590      CALL    ABT
600      EX      DE,HL
610      POP     HL
620      RST     18H
630      RET     NZ
640      LD      DE,-CPL
650      ADD     HL,DE
660      RET
670  ABI      PUSH    HL
680      PUSH    DE
690      PUSH    BC
700      LD      HL,(CURSOR)
710      PUSH    HL
720      CALL    ABT
730      JP      30A7H
740      ;
750  VIDEO    LD      L,(IX+03H)
760      LD      H,(IX+04H)
770      JP      C,317DH
780      LD      A,(IX+05H)
790      OR      A
800      JR      NZ,ABN
810      CALL    CURREN
820      JR      ABO
830  ABN      CALL    CURSET
840  ABO      LD      A,C
850      CP      ' '
860      JR      C,ABS
870      CALL    ABY
880  ABP      JP      3108H
890      ;
900      ; STEUERZEICHEN ERKENNEN
910      ;
920  ABS      LD      DE,ABP
930      PUSH    DE
940      CP      8
950      JP      Z,31DFH
960      CP      0AH
970      RET     C
980      CP      0EH
990      JP      C,ACF
1000     JP      2,31F8H

```

```

1010      CP      0FH
1020      JP      Z,31FDH
1030      CP      18H
1040      RET     C
1050      CP      18H
1060      JP      Z,ACL
1070      CP      19H
1080      JP      Z,ABB
1090      CP      1AH
1100      JP      Z,ACQ
1110      CP      1BH
1120      JP      Z,ACS
1130      CP      1CH
1140      JP      Z,31D4H
1150      CP      1DH
1160      JP      Z,ACH
1170      CP      1EH
1180      JR      Z,ACE
1190      CP      1FH
1200      JR      Z,ACB
1210      RET
1220      ;
1230      ; ZEIGER AUF ZEILENENDE ERMITTELN
1240      ;
1250 ABT      LD      DE,-VIDRAM
1260      LD      B,1
1270      ADD     HL,DE
1280 ABU      LD      DE,CPL
1290      OR      A
1300      SBC     HL,DE
1310      JR      C,ABV
1320      INC     B
1330      JR      ABU
1340 ABV      LD      HL,VIDRAM-CPL
1350 ABW      ADD     HL,DE
1360      DJNZ    ABW
1370      RET
1380      ;
1390      ; ZEICHEN AUF BILDSCHIRM UND SCROLL
1400      ;
1410 ABY      CALL    COLSET
1420      LD      (HL),A
1430      INC     HL
1440      LD      DE,LAST
1450      RST     18H
1460      RET     C
1470      LD      HL,LAST
1480 ABZ      LD      DE,-CPL
1490      ADD     HL,DE
1500      PUSH    HL
1510      EXX
1520      PUSH    BC
1530      PUSH    DE
1540      PUSH    HL
1550      LD      DE,FARBE
1560      LD      HL,FARBE+CPL
1570      LD      BC,GESAMT-CPL
1580      PUSH    BC
1590      EXX
1600      LD      DE,VIDRAM

```

| | | | |
|------|-----|------|-------------------------|
| 1610 | | LD | HL,VIDRAM+CPL |
| 1620 | | POP | BC |
| 1630 | ACA | LDI | |
| 1640 | | EXX | |
| 1650 | | LDI | |
| 1660 | | EXX | |
| 1670 | | LD | A,B |
| 1680 | | OR | C |
| 1690 | | JR | NZ,ACA |
| 1700 | | EXX | |
| 1710 | | POP | HL |
| 1720 | | POP | DE |
| 1730 | | POP | BC |
| 1740 | | EXX | |
| 1750 | | POP | HL |
| 1760 | ACB | LD | DE, LAST |
| 1770 | ACC | JP | 31ACH |
| 1780 | | ; | |
| 1790 | | ; | 1EH |
| 1800 | | ; | |
| 1810 | ACE | PUSH | HL |
| 1820 | | CALL | ABT |
| 1830 | | JP | 31C3H |
| 1840 | | ; | |
| 1850 | | ; | CR |
| 1860 | | ; | |
| 1870 | ACF | CALL | 360AH |
| 1880 | | CALL | ABT |
| 1890 | | ADD | HL,DE |
| 1900 | | LD | DE, LAST |
| 1910 | | RST | 18H |
| 1920 | | JR | Z, ABZ |
| 1930 | | RET | |
| 1940 | | ; | |
| 1950 | | ; | CURSOR ZUM ZEILENANFANG |
| 1960 | | ; | |
| 1970 | ACH | CALL | ABT |
| 1980 | ACI | JP | CURSET |
| 1990 | ACL | PUSH | HL |
| 2000 | | CALL | ABT |
| 2010 | | EX | DE, HL |
| 2020 | | POP | HL |
| 2030 | | RST | 18H |
| 2040 | | JR | Z, ACM |
| 2050 | | DEC | HL |
| 2060 | | JP | CURSET |
| 2070 | ACM | DEC | HL |
| 2080 | | LD | DE, CPL |
| 2090 | | ADD | HL, DE |
| 2100 | | JP | CURSET |
| 2110 | | ; | |
| 2120 | | ; | CURSOR DOWN |
| 2130 | | ; | |
| 2140 | ACQ | LD | DE, CPL |
| 2150 | | CALL | 360AH |
| 2160 | | ADD | HL, DE |
| 2170 | | OR | A |
| 2180 | | CCF | |
| 2190 | | LD | DE, LAST |
| 2200 | | RST | 18H |


```
2210      JR      C,ACR
2220      LD      DE,-GESAMT+CPL
2230      ADD     HL,DE
2240 ACR      JP      CURSET
2250      ;
2260      ; CURSOR UP
2270      ;
2280 ACS      LD      DE,-CPL
2290      CALL    360AH
2300      ADD     HL,DE
2310      LD      DE,VIDRAM
2320      RST     18H
2330      JR      NC,ACT
2340      LD      DE,GESAMT
2350      ADD     HL,DE
2360 ACT      JP      CURSET
2370      ;
2380 RST18N   EX      (SP),HL
2390      LD      A,L
2400      CP      0D8H      ; AUFRUF VON 30D8H
2410      LD      A,H
2420      EX      (SP),HL
2430      JR      NZ,RST18C
2440      CP      30H
2450      JR      NZ,RST18C
2460      LD      HL,GESAMT-1
2470 RST18C   JP      RST18C
2480      ;
2490      END      INIT
```

Kapitel 2 : Die Programmierung der Tonausgabe

Der PSG (Programmable Sound Generator) im Colour-Genie bietet vielfältige Möglichkeiten der Tonerzeugung mit geringem Aufwand. Er hat drei Kanäle, die völlig unabhängig voneinander in Tonhöhe, Lautstärke, Hüllkurve, Rauschen, ... programmiert werden können. Außerdem bietet er 4 parallele Ein-/Ausgabeports.

Die vielen Möglichkeiten des PSG werden über 16 Register programmiert. Für diese 16 Register werden aber nur 2 Portadressen belegt. Daher muß zum Schreiben oder Lesen eines Registers zunächst dessen Nummer in Port E8H (=248) geschrieben werden, wonach der Wert in Port E9H (=249) geschrieben bzw. gelesen werden kann.

Die 16 Register des PSG haben folgende Funktion:

| | | |
|--------|---------|--|
| Reg 0 | Bit 0-7 | : Kanal A Schwingungsdauer Feineinstellung |
| Reg 1 | Bit 0-3 | : Kanal A Schwingungsdauer Grobeinstellung |
| Reg 2 | Bit 0-7 | : Kanal B Schwingungsdauer Feineinstellung |
| Reg 3 | Bit 0-3 | : Kanal B Schwingungsdauer Grobeinstellung |
| Reg 4 | Bit 0-7 | : Kanal C Schwingungsdauer Feineinstellung |
| Reg 5 | Bit 0-3 | : Kanal C Schwingungsdauer Grobeinstellung |
| Reg 6 | Bit 0-4 | : Rauschperiode |
| Reg 7 | Bit 0 | : = 0 Tonausgabe Kanal A ein = 1 Tonausgabe Kanal A aus |
| | Bit 1 | : = 0 Tonausgabe Kanal B ein = 1 Tonausgabe Kanal B aus |
| | Bit 2 | : = 0 Tonausgabe Kanal C ein = 1 Tonausgabe Kanal C aus |
| | Bit 3 | : = 0 Rauschen Kanal A ein = 1 Rauschen Kanal A aus |
| | Bit 4 | : = 0 Rauschen Kanal B ein = 1 Rauschen Kanal B aus |
| | Bit 5 | : = 0 Rauschen Kanal C ein = 1 Rauschen Kanal C aus |
| | Bit 6 | : = 0 I/O Port A Eingabe = 1 I/O Port A Ausgabe |
| | Bit 7 | : = 0 I/O Port B Eingabe = 1 I/O Port B Ausgabe |
| Reg 8 | Bit 0-3 | : Lautstärke Kanal A |
| | Bit 4 | : = 0 Hüllkurve Kanal A aus = 1 Hüllkurve Kanal A ein |
| Reg 9 | Bit 0-3 | : Lautstärke Kanal B |
| | Bit 4 | : = 0 Hüllkurve Kanal B aus = 1 Hüllkurve Kanal B ein |
| Reg 10 | Bit 0-3 | : Lautstärke Kanal C |
| | Bit 4 | : = 0 Hüllkurve Kanal C aus = 1 Hüllkurve Kanal C ein |
| Reg 11 | Bit 0-7 | : Periode der Hüllkurve Feineinstellung |
| Reg 12 | Bit 0-7 | : Periode der Hüllkurve Grobeinstellung |
| Reg 13 | Bit 0 | : HOLD = 0 : Lautstärke am Ende auf 0 |

```

          = 1 : Lautstärke nach einer Periode halten
Bit 1    : ALT
          = 0 : Sprung von 15 auf 0 oder umgekehrt
          = 1 : keine Sprünge
Bit 2    : ATT
          = 0 : mit Lautstärke 15 beginnen
          = 1 : mit Lautstärke 0 beginnen
Bit 3    : CONT
          = 0 : nur eine Periode
          = 1 : beliebig lange fortsetzen
Reg 14 Bit 0-7 : I/O-Port A Daten
Reg 15 Bit 0-7 : I/O-Port B Daten

```

Beispiele:

Tonhöhe:

Die Tonhöhe muß durch einen Teilungsfaktor angegeben werden. Dieser gibt an wie der Haupttakt geteilt werden soll. Dieser Faktor ist 12 Bit lang. Die niederwertigen 8 Bit müssen in Reg 0, 2 oder 4 (Kanal A, B oder C) und die höherwertigen 4 Bit in Reg 1, 3 oder 5 geschrieben werden.

Im Colour-Genie ROM ist eine Tabelle für alle 12 Töne der tiefsten Oktave (1) jeweils in der unsauberen EACA-Stimmung und in einer besseren NCW-Stimmung enthalten:

| | C | D | E | F | G | A | H | CIS | DIS | FIS | GIS | B |
|-------|------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| EACA: | D5B | BE7 | A9B | A02 | 8EB | 7F2 | 714 | C9C | B3C | 973 | 86B | 780 |
| NCW: | 1090 | EC0 | D24 | C68 | B0C | 9D8 | 8C8 | FA0 | DEB | BB4 | A70 | 948 |

Die Tabellen beginnen im ROM bei 3DD1H bzw. 3DF1H. Die 12 Bit langen Schwingungsdauern sind in zwei Byte - zunächst die Grobeinstellung und folgend die Feineinstellung - abgelegt.

So wird ein D der Oktave 1 auf Kanal A programmiert.

```

LD      A,0                ; Reg 0 anwählen
OUT     (0E8H),A
LD      A,0C0H              ; Feineinstellung A
OUT     (0E9H),A           ; programmieren
LD      A,1                ; Reg 1 anwählen
OUT     (0E8H),A
LD      A,0EH              ; Grobeinstellung A
OUT     (0E9H),A           ; programmieren

```

Soll das D nicht in der Oktave 1 sondern eine Oktave höher erklingen, muß statt des Teilungsfaktors 0EC0H der Faktor 760H programmiert werden. Eine Halbierung des Teilungsfaktors bewirkt eine Verdopplung der Frequenz.

Lautstärke:

Um das Rauschen abzuschalten und die Lautstärke auf 15 zu stellen, muß noch folgende Sequenz programmiert werden und der Ton erklingt:

```

LD      A,7           ; Reg 7 anwählen
OUT     (0E8H),A
LD      A,38H         ; = 00111000B
LD      (0E9H),A      ; Rauschen aus, Ton an
;
LD      A,8           ; Reg 8 anwählen
OUT     (0E8H),A
LD      A,15          ; Lautstärke Kanal A
OUT     (0E9H),A      ; = 15

```

Zur Programmierung des PSG können leider nur wenige ROM-Routinen verwendet werden. Die Routine zum PLAY-Befehl kann nicht verwendet werden, da Auswertung der Parameter aus dem Basictext und Verarbeitung zu eng verwoben sind.

```

*****
*
* SOUND : Wert auf PSG-Register ausgeben
*
*      E : 3E32H
*      R : A=E
*      I : A = Registernummer
*      E = Wert
*
*****

```

Funktion:

entspricht BASIC: SOUND A,E

Beispiel:

```

LD      A,8           ; Kanal A Lautstärke
LD      E,15          ; = 15
CALL    SOUND         ; SOUND 8,15

```

```

*****
*
* SNDDE : Wert auf 2 PSG-Register ausgeben
*
*      E : 3E2AH
*      R : A=E
*      I : A = Registernummer
*      DE = Wert
*
*****

```

Funktion:

entspricht BASIC: SOUND A,D:SOUND A-1,E

Beispiel:

```

LD      DE,0EC0H      ; Schwingungsdauer D
LD      A,1           ; Kanal A
CALL    SNDDE

```

Folgendes Beispielprogramm gibt eine Tonleiter über alle Oktaven mit voller Lautstärke aus. Dazu wird aus der Tabelle der Schwingungsdauern der Wert, für den jeweiligen Ton entnommen, und für jede Oktave durch 2 dividiert, und der sich so ergebende Wert in den PSG programmiert.

```

100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* TONLEITER UEBER ALLE OKTAVEN AUSGEBEN
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200 SOUND EQU 3E32H ; SOUND A,E
210 SNDDE EQU 3E2AH ; SOUND A,D:SOUND A-1,E
220 FGR EQU 38A9H
230 CDUR EQU 3DEFH ; TABELLE C-DUR TONLEITER
240 DELAY EQU 0060H ; WARTESCHLEIFE
250 ;
260 ; PROGRAMMSTART
270 ;
280 START LD E,38H ; KEIN RAUSCHEN
290 LD A,7 ; AUF KANAL A-C
300 CALL SOUND
310 LD E,15 ; LAUTSTAERKE 0 = 15
320 LD A,8
330 CALL SOUND
340 LD B,1 ; OKTAVE = 1
350 OKTAVE LD HL,CDUR+2 ; ZEIGER AUF C
360 LD C,1 ; C=TONNR.
370 TON LD E,(HL) ; SCHW.D. LADEN
380 INC HL
390 LD D,(HL)
400 INC HL
410 RLC E ; DE * 2
420 RL D
430 PUSH BC ; OKTAVE RETTEN
440 DIV SRL D ; DE / 2, PRO OKTAVE
450 RR E
460 DJNZ DIV
470 LD BC,40000 ; WARTEN
480 CALL DELAY
490 LD A,1 ; TON AUF KANAL A
500 CALL SNDDE
510 POP BC ; OKTAVE ZURUECK
520 INC C ; 7 TOENE
530 LD A,C
540 CP 8
550 JR NZ,TON ; NEIN --> NAECHSTER TON
560 INC B ; 8 OKTAVEN
570 LD A,B
580 CP 9
590 JR NZ,OKTAVE ; NEIN --> NAECHSTER TON
600 RET
610 END START

```

1.2.1 Die Programmierung der Parallelschnittstelle Seite 38

2.1 Die Programmierung der Parallelschnittstelle

Eine Parallelschnittstelle ist im PSG eingebaut. Sie besteht aus 2 Port mit je 8 Bits. Jeder Port kann zur Eingabe oder Ausgabe benutzt werden. Dies wird mit den Bits 6 und 7 im Register 7 festgelegt.

Die Ports selbst werden über die Register 14 und 15 gesetzt bzw. gelesen.

Beispiel:

```
LD      A,7                ; Reg 7 anwählen
OUT     (0E8H),A
LD      A,40H              ; Port A Ausgabe
OUT     (0E9H),A          ; Port B Eingabe
;
LD      A,14               ; Port A anwählen
OUT     (0E8H),A
LD      A,0FFH             ; Alle Bits setzen
OUT     (0E9H),A
;
LD      A,15               ; Port B anwählen
OUT     (0E8H),A
IN      A,(0E9H)           ; Port B abfragen
```

Die Centronics-Parallelschnittstelle für den Drucker wird über die Ports A und B des PSG bedient.

| | | |
|----------------|---|----------------------|
| Port A Bit 0-7 | : | Ausgabe des Zeichens |
| Port B Bit 0 | : | Ausgabe des Strobe |
| Port B Bit 7 | : | Abfrage Busy |
| Bit 6 | : | Out of Paper |
| Bit 5 | : | Unit Select |
| Bit 4 | : | Ready |

Der Strobe wird durch ein Monoflop auf dem Druckerinterface verlängert. Für etwa die 10 fache Zeitdauer wird automatisch ein Busy generiert, auch wenn der Drucker dies nicht anzeigt.

Kapitel 3 : Der Betrieb mit der Diskettenstation

1.3.1 Allgemeines

Für den Zugriff auf Diskettenfiles stellt das DOS im ROM eine ganze Anzahl leistungsfähiger Unterprogramme zur Verfügung, die größtenteils den entsprechenden Unterprogrammen im TRSDOS, NEWDOS und GDOS entsprechen. Lediglich die Einsprungadressen unterscheiden sich. Während bei den genannten Betriebssystemen alle Einsprünge in der Page von 4400H-44FFH liegen, sind sie beim Colour-Genie in die Page CE00H-CEFFH verlegt. Die LSB's der Adressen stimmen aber überein. Dem CALL 4428H unter GDOS, ... entspricht also der CALL 0CE28H auf dem Colour-Genie. Beide Unterprogrammaufrufe bewirken das Schließen eines FCB's.

Der File Control Block

Der File Control Block (FCB) enthält alle Informationen über ein geöffnetes File, die das DOS zum Zugriff benötigt. Dieser 32 Byte lange Speicherbereich bildet die Verbindung zwischen dem Programm und dem DOS. Beim Eröffnen eines Files mittels INIT oder OPEN müssen die ersten Bytes den mit CR abgeschlossenen Filenamen enthalten. Diese Routinen füllen den FCB dann mit Daten von der Diskette, wodurch das File dann mit den anderen Unterprogrammen bearbeitet werden kann. Durch die CLOSE-Routine wird der Filename wieder in den FCB eingeschrieben.

Ein FCB zu einem geöffneten File ist folgendermaßen aufgebaut:

FCB : Bit 7 = 1 : File ist eröffnet
 = 0 : File ist geschlossen
 Bit 1 = 1 : NEXT und EOF werden Disk-relativ interpretiert
 Bit 0 = 1 : Sektoren werden lesegeschützt geschrieben. Fehler 6 beim Lesen!
FCB + 3 : LSB der Bufferadresse, die beim OPEN in HL stand
FCB + 4 : MSB
FCB + 5 : NEXT BYTE, Nr. des Bytes im Sektor (FCB + 10,11), welches zunächst verarbeitet wird.
FCB + 6 : Laufwerksnummer
FCB + 8 : EOF BYTE, Nr. des EOF-Byte im Sektor (FCB +12,13)
FCB + 10: LSB NEXT, des Sektor der zunächst gelesen oder geschrieben wird.
FCB + 11: MSB NEXT
FCB + 12: LSB EOF, LSB des letzten Filesektors
FCB + 13: MSB EOF

Besonders wichtig im FCB sind die NEXT und EOF-Felder, die jeweils 3 Byte belegen. Das 2. und 3. Byte (MSB + LSB) geben die Sektornummer an. Das 1. Byte gibt die Nummer des Bytes im Sektor an. Beim Sektorzugriff ist die Nummer des Bytes unerheblich.

Die weiteren Bytes des FCB werden mit Daten aus dem Directory-Entry des Files gefüllt und sind hier nicht von besonderem Interesse. Genauere Informationen dazu können dem NEWDOS80 V2.0 Handbuch entnommen werden.

Beim Zugriff auf ein File mit den beschriebenen Unterprogrammen muß man sich um den Aufbau des FCB nicht kümmern. Beim Eröffnen eines Files mittels OPEN oder INIT wird er automatisch eingerichtet.

Für viele DOS-Unterprogramme gelten ähnliche Parameter. In DE wird die Adresse des FCB übergeben, der, wenn nichts anderes angegeben ist, eröffnet sein muß. Bei der Rückkehr zeigt das gesetzte Zero-Flag erfolgreiche Bearbeitung des Unterprogrammes an. Andernfalls ist im Akku die Fehlernummer, deren Bedeutung im DOS-Handbuch nachgelesen werden kann.

1.3.2 Eröffnen von Files

```

*****
*
* INIT   : evtl. nicht existierendes File eröffnen
*
*       E : CE20H
*       R : AF
*       I : DE = Zeiger auf Filenamen
*           (DE) = 1. Zeichen des Filenamen
*           (DE+n) = 0DH
*           B = logische Recordlänge
*           HL = Bufferadresse
*       O : Z = 0 : kein Fehler
*           Z = 1 : Fehler
*           A : Fehlernummer
*
*****

```

Funktion:

DE weist auf einen Filenamen, der nach den Richtlinien für Filenamen aufgebaut sein muß. Er kann also nach einem Schrägstrich eine Erweiterung, nach einem Doppelpunkt eine Laufwerksbezeichnung oder nach einem Punkt ein Password enthalten. Existiert ein File mit dem angegebenen Namen, so wird es eröffnet. Anderfalls wird ein File neu eingerichtet.

Beispiel:

```

LD      DE,FCB          ; FCB-Adresse nach DE
LD      HL,BUFFER       ; Bufferadresse nach HL
LD      B,0             ; Logische Sektorlänge =256
CALL    INIT            ; Neues File erzeugen
JR      NZ,...          ; Fehler, -->
;
FCB     DEFM 'TEST/TXT:1'
        DEFB 0DH
        DEFS 32 - $ + FCB ; 32 Bytes für FCB !
BUFFER  DEFS 256         ; 256 für Buffer
                        ; auch wenn B <> 0

```

```
*****
*
* OPEN   : existierendes File eröffnen
*
*       E : CE24H
*       R : AF
*       I : DE = Zeiger auf Filenamen
*           (DE) = 1. Zeichen des Filenamen
*           (DE+n) = 0DH
*           B = logische Recordlänge
*           HL = Bufferadresse
*       O : Z = 0 : kein Fehler
*           Z = 1 : Fehler
*           A : Fehlernummer
*
*****
```

Funktion:

Wie OPEN. Existiert das File aber noch nicht, wird mit Fehler abgebrochen.

Beispiel:

siehe INIT

1.3.3 Schreiben und Lesen von Files

Byteweises Schreiben:

```

*****
*
* RDBYTE : Byte von File lesen
*
*     E : 0013H
*     R : AF
*     I : DE = Zeiger auf eröffneten FCB
*     O : Z = 0 : kein Fehler
*           A : gelesenes Byte
*     Z = 1 : Fehler
*           A : Fehlernummer
*
*****

```

Funktion:

Das nächste Byte des Files wird gelesen, wobei bei Bedarf ein Sektor in den Buffer nachgeladen wird. Der Einsprung für diese Routine liegt in der Page 0! über RAM-Vektoren wird zur ausführenden Routine im DOS verzweigt.

Beispiel:

```

      LD      DE,FCB          ; Zeiger auf geöffneten FCB
LOOP  CALL    RDBYTE          ; 1 Byte lesen
      JR      NZ,ERROR        ; Fehler ? -->
      PUSH    DE
      CALL    OUTCH           ; Zeichen ausgeben
      POP     DE
      JR      LOOP            ; weiter
ERROR  CP      28              ; Ende der Datei ?
      RET     Z                ; ja, fertig
      ; Fehlerbearbeitung

```

```

*****
*
* WRBYTE : Byte auf File schreiben
*
*     E : 001BH
*     R : AF
*     I : DE = Zeiger auf eröffneten FCB
*           A = zu schreibendes Byte
*     O : Z = 0 : kein Fehler
*           Z = 1 : Fehler
*           A : Fehlernummer
*
*****

```

Funktion:

Das Byte im Akku wird auf das File geschrieben, wobei falls

erforderlich ein Sektor auf Diskette geschrieben wird.

Beispiel:

```

      LD      DE,FCB          ; DE zeigt auf eröffn. FCB
LOOP  LD      A,(HL)          ; Zeichen laden
      AND     A              ; Ende ?
      JR      Z,...          ; ja, -->
      CALL    WRBYTE         ; Byte schreiben
      JR      NZ,...         ; Fehler, -->
      INC     HL
      JR      LOOP          ; weiter

```

Sektoren schreiben und lesen:

```

*****
*
* RDSEC  : Sektor von File lesen
*
*      E : CE36H
*      R : AF
*      I : DE = Zeiger auf eröffneten FCB
*          B <> 0 beim Eröffnen per OPEN:
*          HL = Zeiger auf Buffer der Länge B (OPEN)
*      O : B <> 0 beim Eröffnen per OPEN:
*          (HL)...(HL+B-1) : gelesener Record
*          Andernfalls:
*          gelesener Sektor im Sektorbuffer
*          (HL bei OPEN)
*          Z = 0 : kein Fehler
*          Z = 1 : Fehler
*          A : Fehlernummer
*
*****

```

Funktion:

Beim Lesen von Sektoren muß zwischen Files mit logischen Sektorlängen von 256 und anderen unterschieden werden. Wurde ein File mit einem Wert von $B \neq 0$ eröffnet, dann muß beim Aufruf von RDSEC in HL die Adresse eines Buffer entsprechender Länge übergeben werden, in den durch dieses Unterprogramm die nächsten B Bytes des Files übertragen werden. Wird dadurch eine phys. Sektorgrenze überschritten, wird von Diskette der nächste physikalische Sektor in den Sektorbuffer (HL beim Öffnen des Files per OPEN) geladen. Wurde aber das File mit der Sektorlänge 256 eröffnet, dann muß beim Aufruf von RDSEC nichts in HL übergeben werden, und es wird nur der nächste Sektor von der Diskette in den Sektorbuffer geladen.

Mit der RDSEC-Routine ist auch das Lesen von Disk-relativ adressierten Sektoren möglich. Dazu muß das Bit 1 von (FCB) gesetzt sein. In diesem Fall werden die NEXT und EOF-Felder des FCB nicht File-relativ sondern Disk-relativ interpretiert.

Wird der FCB mit $B=0$ eröffnet, ist für die weiteren Zugriffe (FCB+5) unwichtig. (FCB+10) und (FCB+11) geben jeweils die Nummer des nächsten Sektors an.

Beispiel:

```
LD      DE,FCB          ; File eröffnen
LD      HL,BUFFER
LD      B,128           ; logische Recordlänge= 128
CALL    OPEN
JR      NZ,FEHLER       ; Fehler, -->
LD      HL,RECBUF       ; logischen Record -> (HL)
CALL    RDSEC
JR      NZ,FEHLER       ; Fehler, -->
...
FCB      DEFM 'TEST/TXT:0'
          DEFB 0DH
          ORG  FCB+32
BUFFER   DEFS 256        ; Sektorbuffer für OPEN
RECBUF   DEFS 128        ; Buffer für log. Record
```

```

*****
*
* WRSEC : Sektor von File lesen
*
*      E : CE39H
*      R : AF
*      I : DE = Zeiger auf eröffneten FCB
*          B <> 0 beim Eröffnen per OPEN:
*          HL = Zeiger auf Buffer der Länge B (OPEN)
*          (HL)...(HL+B-1) : zu schreibender Record
*      Andernfalls:
*          zu schreibender Sektor im Sektorbuffer
*          (HL bei OPEN)
*      O : Z = 0 : kein Fehler
*          Z = 1 : Fehler
*          A : Fehlernummer
*
*****

```

Funktion:

Beim Schreiben von Sektoren muß genau wie beim Lesen zwischen Sektoroperationen und logischen Recordoperationen unterschieden werden. Ist B beim Öffnen des Files mit OPEN 0, wird der Inhalt des Buffer, dessen Adresse beim OPEN in HL angegeben wird, auf das File geschrieben (256 Bytes). Andernfalls werden die B Bytes, die in dem Speicherbereich, auf den HL beim Aufruf von WRSEC weist, in den Sektorbuffer übertragen, und - falls erforderlich - dieser auf das File geschrieben.

Ist das Bit 0 von (FCB) gesetzt, wird der Sektor lesegeschützt geschrieben. Derartig geschriebene Sektoren ergeben beim Lesen einen Fehler 6. Das DOS schreibt den Directory lesegeschützt.

Wie bei RDSEC ist auch hier Disk-relatives Schreiben möglich.

Beispiel:

```

LD      DE,FCB          ; File eröffnen
LD      HL,BUFFER
LD      B,0             ; Sektoroperation
CALL    OPEN
JR      NZ,FEHLER       ; Fehler ? -->
LD      HL,DATEN        ; Sektorbuffer füllen
LD      DE,BUFFER
LD      BC,256
LDIR
LD      DE,FCB          ; Sektorbuffer schreiben
CALL    WRSEC
JR      NZ,FEHLER       ; Fehler, -->
...
FCB     DEFM 'TEST/TXT:0'
        DEFB 0DH
        ORG  FCB+32
BUFFER  DEFS 256         ; Sektorbuffer für OPEN
DATEN   DEFS 256         ; Speicherbereich in dem
                           ; Daten stehen

```

1.3.4 Positionierung des NEXT

Mit den folgenden Routinen kann das NEXT-Feld im FCB geändert werden und damit auf einen speziellen Sektor im File zugegriffen werden, ohne das File vom Beginn an sequentiell lesen.

```
*****
*
* POS0    : NEXT-Feld auf Sektor 0 positionieren
*
*      E : CE3FH
*      R : AF
*      I : DE = Zeiger auf eröffneten FCB
*      O : Z = 0 : kein Fehler
*          Z = 1 : Fehler
*          A : Fehlernummer
*
*****
```

Funktion:

Das NEXT-Feld im FCB wird auf den Fileanfang gesetzt. Ist im Buffer ein Sektor, der noch auf die Diskette geschrieben werden muß, wird dieser zunächst auf die Diskette geschrieben.

Beispiel:

```
LD      DE,FCB          ; Zeiger auf geöffneten FCB
CALL    POS0            ; FCB auf Sektor 0 pos.
CALL    RDSEC           ; Sektor 0 laden
```

```
*****
*
* POSBC   : NEXT-Feld im FCB auf BC positionieren
*
*      E : CE42H
*      R : AF
*      I : DE = Zeiger auf eröffneten FCB
*          BC = logische Recordnummer
*      O : Z = 0 : kein Fehler
*          Z = 1 : Fehler
*          A : Fehlernummer
*
*****
```

Funktion:

Das NEXT-Feld im FCB wird gleich BC * logische Recordlänge gesetzt. Mit dem nächsten Filezugriff wird also der logische Record BC angesprochen. Ansonsten gleiche Funktion wie POS0.

Beispiel:

```
LD      DE,FCB          ; Zeiger auf geöffneten FCB
LD      BC,17           ; Recordnummer = 17
CALL    POS0            ; FCB auf Record 17 pos.
LD      HL,LOGBUF        ; Zeiger auf log. RecBuffer
CALL    RDSEC           ; Record 17 laden
```

```
*****
*
* POSDEC : NEXT-Feld um einen Record decrementieren
*
*      E : CE45H
*      R : AF
*      I : DE = Zeiger auf eröffneten FCB
*      O : Z = 0 : kein Fehler
*          Z = 1 : Fehler
*          A : Fehlernummer
*
*****
```

Funktion:

Das NEXT-Feld im FCB wird um die logische Recordlänge erniedrigt. Ansonsten gleiche Funktion wie POS0.

Beispiel:

```
LD      DE,FCB          ; Zeiger auf geöffneten FCB
LD      HL,RECBUF        ; Record-Buffer
CALL    RDSEC            ; log. Record lesen
CALL    POSDEC           ; FCB auf einen Record zur.
CALL    RDSEC            ; gleichen Record noch mal
```

```
*****
*
* POSEOF : NEXT-Feld auf EOF positionieren
*
*      E : CE48H
*      R : AF
*      I : DE = Zeiger auf eröffneten FCB
*      O : Z = 0 : kein Fehler
*          Z = 1 : Fehler
*          A : Fehlernummer
*
*****
```

Funktion:

Das NEXT-Feld im FCB wird auf das Fileende gesetzt. Ansonsten siehe POS0.

Beispiel:

```
LD      DE,FCB          ; Zeiger auf geöffneten FCB
CALL    POS0            ; FCB auf EOF pos.
CALL    WRSEC           ; File erweitern
```



```

*****
*
* POSRBA : NEXT-Feld auf HLC positionieren
*
*      E : CE4EH
*      R : AF
*      I : DE = Zeiger auf eröffneten FCB
*          HL = NEXT (H = MSB, L = LSB)
*          C = NEXT Byte
*      O : Z = 0 : kein Fehler
*          Z = 1 : Fehler
*          A : Fehlernummer
*
*****

```

Funktion:

Das NEXT-Feld im FCB wird auf HL und C gesetzt. HL gibt die Sektornummer an und C die Position im Record. Funktion sonst wie POS0.

Beispiel:

```

LD      DE,FCB          ; Zeiger auf geöffneten FCB
LD      HL,28           ; Sektornummer = 28
LD      C,80H           ; Byte = 128
CALL    POS0            ; FCB auf Sektor 0 pos.
LD      HL,SECBUF        ; logischen Record lesen
CALL    RDSEC

```

1.3.5 File Laden, Starten, Löschen

```
*****
*
* LOAD      : File im Loaderformat laden
*
*      E : CE30H
*      R : AF
*      I : DE = Zeiger auf Filenamen
*           (DE) = 1. Zeichen des Filenamen
*           (DE+n) = 0DH
*      O : Z = 0 : kein Fehler
*           Z = 1 : Fehler
*           A : Fehlernummer
*
*****
```

Funktion:

Ein File im Loaderformat wird geladen. Die Speicheradressen, an die es geladen wird, sind im Loaderformat enthalten, das folgendermaßen aufgebaut ist:

1. Datenblock:

- a. Kennung = 01H
- b. Anzahl der Datenbytes + 2 (0 - 2 = 256 - 258)
- c. Ladeadresse (LSB/MSB)
- d. Datenbytes

2. Startadressenblock:

- a. Kennung = 02H
- b. Anzahl der folgenden Byte (immer 02H)
- c. Startadresse (LSB/MSB)

3. Kommentarblock

- a. Kennung = 0 oder 03H-FFH
- b. Anzahl der folgenden Kommentarbytes (0 = 256)
- c. Kommentarbytes

Beispiel:

CMD"W DEMO,0,300,0"

1. Datenblock: 01 00 00 00 F3 AF .. (250 Bytes) .. 14 44
 2. Datenblock: 01 31 FE 00 DF DA .. (43 Bytes) .. 69 63
 Startadrblock: 02 02 00 00

Beispiel:

```
LD      DE,FCB          ; Zeiger auf Filename
CALL    LOAD            ; File laden
;
FCB     DEFS             'DEMO'
```

```

DEFB      13
; Dies provoziert übrigens einen Fehler 36

```

```

*****
*
* RUN      : File im Loaderformat laden und starten
*
*      E : CE33H
*      R : AF
*      I : DE = Zeiger auf Filenamen
*           (DE) = 1. Zeichen des Filenamen
*           (DE+n) = 0DH
*      O : Z = 0 : kein Fehler
*           Z = 1 : Fehler
*           A : Fehlernummer
*
*****

```

Funktion:

Ein File im Loaderformat wird geladen und gestartet.

Beispiel:

```

LD      DE,FCB      ; Zeiger auf Filenamen
CALL    RUN          ; File laden und starten
...
FCB      DEFM      'COLASM/CMD'
DEFB     13

```

```

*****
*
* KILL     : File löschen
*
*      E : CE2CH
*      R : AF
*      I : DE = Zeiger auf Filenamen
*           (DE) = 1. Zeichen des Filenamen
*           (DE+n) = 0DH
*      O : Z = 0 : kein Fehler
*           Z = 1 : Fehler
*           A : Fehlernummer
*
*****

```

Funktion:

Das File, dessen Namen im FCB steht, wird gelöscht.

Beispiel:

```

LD      DE,FCB      ; Zeiger auf Filenamen
CALL    KILL         ; File löschen
JP      NZ,FEHLER    ; Fehler ? -->
...
FCB      DEFM      'TEST'
DEFB     13

```

3.6 File schließen

```

*****
*
* CLOSE  : File löschen
*
*      E : CE28H
*      R : AF
*      I : DE = Zeiger auf eröffneten FCB
*      O : Z = 0 : kein Fehler
*          Z = 1 : Fehler
*          A : Fehlernummer
*      (DE) = 1. Zeichen des Filenamens
*      (DE+n) = 0DH
*
*****

```

Funktion:

Das dem geöffneten FCB zugehörige File, wird geschlossen. Dazu wird evtl. der letzte Sektor auf Diskette und ein neuer EOF auf Diskette geschrieben und nicht mehr benutzte Grans werden wieder freigegeben. Schließlich wird der Filename wieder in der FCB geschrieben.

Beispiel:

```

LD      DE,FCB      ; Zeiger auf geöffneten FCB
CALL    KILL        ; File schließen

```

1.3.7 Beispiele zum Diskettenbetrieb

Das erste Beispielprogramm ermöglicht die Ausgabe eines beliebigen Files auf den Bildschirm. Der Filename wird abgefragt, das File eröffnet und dann zeichenweise gelesen und auf den Bildschirm ausgegeben.

```

100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* LESEN EINES FILES
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200 OPEN      EQU      0CE24H  ; FILE OEFFNEN
210 RDBYTE    EQU      0013H  ; BYTE LESEN
220 INBUFF    EQU      05D9H  ; EINGABE EINER ZEILE
230 PRINTT    EQU      2B75H  ; TEXT AUSGEBEN
240 OUTCH      EQU      0033H  ; ZEICHEN AUSGEBEN
250 START     LD        HL,TEXT ; "FILENAME ?" AUSGEBEN
260           CALL      PRINTT
270           LD        HL,FCB  ; FILENAME IN FCB
280           LD        B,20
290           CALL      INBUFF  ; EINGEBEN
300           JR        C,START ; BREAK -->
310           EX        DE,HL   ; ZEIGER AUF FCB NACH DE
320           LD        HL,BUFFER
330           CALL      OPEN    ; FILE EROEFFNEN
340           JR        NZ,FEHLER ; FEHLER -->
350 LOOP      CALL      RDBYTE  ; BYTE LESEN
360           JR        NZ,FEHLER ; FEHLER -->
370           PUSH      DE      ; FCB-ADR. RETTEN
380           CALL      OUTCH   ; AUSGEBEN
390           POP       DE
400           JR        LOOP    ; WEITER
410 ENDE      CP        28      ; ENDE DES FILES
420           JR        Z,START ; JA, NAECHSTES FILE
430 FEHLER    LD        HL,FEHMEL ; FEHLERMELDUNG AUSGEBEN
440           CALL      PRINTT
450           JR        START   ; NEU BEGINN
460           ;
470           ; TEXTE
480           ;
490 TEXT      DEFM      'FILENAME ?'
500           DEFB      0
510 FEHMEL    DEFM      'DOSFEHLER !'
520           DEFB      13
530           DEFB      0
540           ;
550           ; BUFFER
560           ;
570 FCB        DEFS      32
580 BUFFER     DEFS      256
590           END        START

```

Folgendes Beispiel demonstriert den disk-relativen Zugriff auf einzelne Sektoren. Die Sektornummer wird abgefragt und in das NEXT-Feld des FCB eingetragen. Weiterhin ist wichtig, daß im FCB der EOF auf den Wert 65535 gesetzt wird und das Bit für disk-relativen Zugriff gesetzt wird. Der gelesene Sektor wird in 16 Zeilen a 16 Spalten auf dem Bildschirm ausgegeben. Der Direktory beginnt im Sektor 360.

```

100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* DISKRELATIVEN SEKTOR LESEN
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200 RDSEC EQU 0CE36H ; SEKTOR LESEN
210 INBUFF EQU 05D9H ; EINGABE IM BUFFER
220 DECLNO EQU 1E4FH ; STRING IN ZAHL
230 PRINTT EQU 2B75H ; STRING AUSGEBEN
240 OUTCH EQU 0033H ; ZEICHEN AUSGEBEN
250 START LD HL,TEXT1 ; FRAGE AUSGEBEN
260 CALL PRINTT
270 LD HL,BUFFER
280 LD B,5
290 CALL INBUFF ; ANTWORT LESEN
300 JR C,START ; BREAK -->
310 CALL DECLNO ; SEKTORNR. DEC.
320 LD (FCB+10),DE ; IN NEXT LSB/MSB
330 LD DE,FCB
340 CALL RDSEC ; SEKTOR LESEN
350 JR NZ,FEHLER ; FEHLER, -->
360 ;
370 ; SEKTOR AUSGEBEN
380 ;
390 OK LD HL,BUFFER
400 LD BC,16 ; B = 0 (256), C = 16 ZEICHEN /
ZEILE
410 LOOP LD A,(HL) ; ZEICHEN LADEN
420 CP ' '
430 INC HL
440 JR NC,L1 ; >= ' ' -->
450 LD A,' ' ; STEUERCODES DURCH ' ' ERS.
460 L1 CALL OUTCH ; ZEICHEN AUSGEBEN
470 DEC C ; 16 ZEICHEN AUSGEGBEN
480 JR NZ,L2 ; NEIN -->
490 LD C,16
500 LD A,0DH ; CR AUSGEBEN
510 CALL OUTCH
520 L2 DJNZ LOOP ; WEITER
530 JP START
540 ;
550 ; FEHLERBEHANDLUNG
560 ;
570 FEHLER CP 6 ; DIR SEKTOR
580 JR Z,OK ; JA -->

```

```
590          LD      A,'?'      ; ANSONSTEN NEUBEGINN
600          CALL    OUTCH
610          JP      START
620          ;
630          ; TEXTE
640          ;
650 TEXT1     DEFM     'SEKTOR IN LAUFWERK 0 ?'
660          DEFB     0
670 FCB       DEFB     82H      ; FILE OFFEN, DISK RELATIV
680          DEFB     0
690          DEFB     0
700          DEFW     BUFFER    ; BUFFERADRESSE
710          DEFB     0
720          DEFB     0
730          DEFB     0
740          DEFB     0
750          DEFB     0
760          DEFW     0          ; NEXT
770          DEFW     0FFFFH    ; EOF = 65535
780          DEFB     0
790          DEFB     0
800          DEFB     0
810          DEFB     0
820          DEFB     0
830          DEFB     0
840          DEFB     0
850          DEFB     0
860          DEFB     0
870          DEFB     0
880          DEFB     0
890          DEFB     0
900          DEFB     0
910          DEFB     0
920          DEFB     0
930          DEFB     0
940          DEFB     0
950          DEFB     0
960 BUFFER    EQU      *
970          END      START
```

Das nächste Beispiel zeigt die Anwendung der Positionierungsroutinen. Insbesondere beim Arbeiten mit logischen Records können diese Routinen sinnvoll eingesetzt werden. Nach dem Öffnen des Files mit der logischen Recordlänge von 100, werden zunächst 10 logische Records, die jeweils nur aus der Nummer und Leerzeichen bestehen, auf das File geschrieben und dieses wieder geschlossen. Danach wird es wieder geöffnet und einzelne Records unter Zuhilfenahme der verschiedenen Positionierungsroutinen gelesen und auf dem Bildschirm ausgegeben.

Hier noch mal ein Überblick über die verschiedenen Positionierungsroutinen

Folgende Tabelle gibt an, auf welchen logischen Record nach Aufruf der entsprechenden Positionierungsroutine zugegriffen werden kann und welchen Wert das NEXT-Feld im FCB nachher hat.

Alleinige Änderung des FCB bewirkt nicht das gleiche wie die POSx-Calls, da diese gegebenenfalls den momentanen Buffer auf Diskette schreiben.

| | ! | Log. Record. Nr. | NEXT | MSB | LSB | BYTE |
|--------|---|-------------------|----------------------|-----|-----|------|
| | ! | | | | | |
| POS0 | ! | 0 | 00 | 00 | 00 | |
| POSDEC | ! | -1 | - log. Rec. Länge | | | |
| POSBC | ! | BC | BC * log. Rec. Länge | | | |
| POSRBA | ! | RBA / log. Rec.L. | H | L | C | |
| POSEOF | ! | hinter letzten | = EOF | | | |

```

100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* TEST ZUR POSITIONIERUNG
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200 INIT EQU 0CE20H ; FILE OEFFNEN
210 OPEN EQU 0CE24H ; FILE OEFFNEN
220 WRSEC EQU 0CE39H ; REC. SCHREIBEN
230 RDSEC EQU 0CE36H ; REC. LESEN
240 POS0 EQU 0CE3FH ; POS AUF SEKTOR 0
250 POSBC EQU 0CE42H ; POS AUF LOG. REC BC
260 POSDEC EQU 0CE45H ; POS EINEN SEC. ZURUECK
270 POSEOF EQU 0CE48H ; POS AUF EOF
280 POSRBA EQU 0CE4EH ; NEXT = HLC
290 CLOSE EQU 0CE28H ; FILE SCHLIESSEN
300 OUTCH EQU 0033H ; ZEICHEN AUSGEBEN
310 ;
320 START LD B,100 ; LOG. RECLAENGE = 100
330 LD DE,FCB ; ZEIGER AUF FCB NACH DE
340 LD HL,BUFFER
350 CALL INIT ; FILE EROEFFNEN
360 JR NZ,FEHLER ; FEHLER -->

```



```

370      LD      HL,LOGBUF ; LOGBUF MIT BLANK FUELLEN
380      PUSH   HL
390 FLOOP LD      (HL), ' ' ; BLANK IN BUFFER
400      INC     HL
410      DJNZ    FLOOP ; 100 MAL
420      POP     HL
430      LD      B,'0' ; 10 SEC. SCHREIBEN
440 SLOOP LD      (HL),B ; REC.NR IN BUFFER
450      CALL    WRSEC ; LOG. SEKTOR SCHREIBEN
460      JR      NZ,FEHLER
470      INC     B ; REC.NR. + 1
480      LD      A,B ; 10 GESCHRIEBEN
490      CP      '0'+10
500      JR      NZ,SLOOP ; NEIN, WEITER
510      CALL    CLOSE ; FILE SCHLIESSEN
520      JR      NZ,FEHLER ; FEHLER -->
530      LD      B,100 ; LOG. RECLAENGE = 100
540      LD      DE,FCB ; ZEIGER AUF FCB NACH DE
550      LD      HL,BUFFER
560      CALL    OPEN ; FILE EROEFFNEN
570      JR      NZ,FEHLER ; FEHLER -->
580      LD      HL,LOGBUF
590      CALL    RDSEC ; 1. REC LESEN
600      CALL    PRBUF ; UND AUSGEBEN
610      LD      BC,8 ; AUF 8. REC POS
620      CALL    POSBC
630      JR      NZ,FEHLER ; FEHLER -->
640      CALL    RDSEC ; UND LESEN
650      JR      NZ,FEHLER ; FEHLER -->
660      CALL    PRBUF
670      CALL    POSDEC ; REC 8 NOCH MAL
680      JR      NZ,FEHLER ; FEHLER -->
690      CALL    RDSEC ; LESEN
700      JR      NZ,FEHLER ; FEHLER -->
710      CALL    PRBUF
720      LD      HL,1 ; AUF BYTE 500
730      LD      C,500-256
740      CALL    POSRBA ; POS
750      JR      NZ,FEHLER ; FEHLER -->
760      LD      HL,LOGBUF
770      CALL    RDSEC ; REC 5 LESEN
780      JR      NZ,FEHLER ; FEHLER -->
790      CALL    PRBUF
800      CALL    POSEOF ; AUF FILEENDE POS.
810      JR      NZ,FEHLER ; FEHLER -->
820      CALL    RDSEC ; HINTER FILEENDE LESEN640
830      ; GIBT IMMER FEHLER !
830      JR      NZ,FEHLER ; FEHLER -->
840      RET
850      ;
860 FEHLER LD      A,'?' ; '?' FUER FEHLER AUSGEBEN
870      JP      OUTCH
880      ;
890 PRBUF LD      HL,LOGBUF ; BUFFER AUSGEBEN
900      PUSH   HL ; LOGBUF RETTEN
910      LD      B,100 ; LAENGE = 100
920 PLOOP LD      A,(HL) ; ZEICHEN LADEN
930      INC     HL
940      PUSH   DE ; FCB-ADRESSE RETTEN
950      CALL    OUTCH ; UND AUSGEBEN

```

```
960      POP      DE      ; FCB-ADRESSE ZURUECK
970      DJNZ     PLOOP   ; WEITER BIS 100
980      POP      HL      ; LOGBUF ZURUECK
990      RET
1000     ;
1010     ; BUFFER
1020     ;
1030 FCB   DEFM     'TEST' ; FILENAME
1040      DEFB     13      ; CR
1050      DEFS     27      ; 27 + 5 = 32
1060 BUFFER DEFS     256   ; SEKTOTBUFFER
1070 LOGBUF DEFS     100   ; BUFFER FUR LOG. RECORD
1080      END      START
```

Das letzte Beispiel zeigt ein Basicprogramm zur Reparatur eines zerstörten HIT-Sektors. Der HIT-Sektor (der zweite des Directory) enthält die Hashcodes aller Files. Wenn er zerstört ist, kann kein File mehr gelesen werden. Wenn er vollständig zerstört ist, hilft auch folgendes Basicprogramm erst, nachdem auf der Position 2 des HIT-Sektors der Hashcode des Directory (DIR/SYS) C4H eingetragen wurde, da sonst der Directory nicht gefunden wird. Das Unterprogramm ab Zeile 80 ermittelt den Hashcode eines Filenamens, der in A\$ stehen muß. Dieser muß genau 11 Zeichen lang sein, wobei Filename und -erweiterung jeweils mit Leerzeichen aufzufüllen sind.

Die HIT-Sektor wird zunächst in das Feld H(I) eingelesen. Dann werden alle weiteren Directory-Sektoren gelesen und die aktiven Einträge gesucht. Bei diesen muß noch zwischen FPDE's und FXDE's unterschieden werden. Die Ersteinträge (FPDE's) enthalten den Filenamen aus dem der Hashcode ermittelt und in den HIT (H1(I)) eingetragen werden kann. Alle folgende DIR-Einträge eines Files enthalten einen Bezug auf den vorangehenden Eintrag, aus dem der Hashcode kopiert werden kann.

Ist der gesamte HIT in H1(I) neu generiert, muß er wieder in den DIR/SYS auf Diskette geschrieben werden. Da der Directory lesegeschützt auf der Diskette stehen muß, wird zuvor durch den POKE in den FCB von File 1 (Zeile 480) erreicht, das der Schreibbefehl einen Sektor lesegeschützt schreibt.

```

10 CLEAR 1000
20 DEFINT A-Z:DIMHI(255)
30 DIM H(255),H1(255):ON ERROR GOTO 580
40 OPEN"R",1,"DIR/SYS"
50 FIELD 1,128 AS A$,128 AS B$
60 DEF FN EXO (A,B) = (A OR B) AND NOT (A AND B)
70 GOTO 170
80 REM UNTERPROGRAMM HASHCODE
90 HC=0
100 FOR II=1 TO 11
110 HC = FN EXO (ASC(MID$(F$,II,1)),HC)
120 HC = HC * 2
130 IF HC>255 THEN HC=HC-255
140 NEXT
150 IF HC=0 THEN HC=1
160 RETURN
170 REM HAUPTPROGRAMM
180 GET 1,2
190 FOR I=0 TO 255:H1(I)=0:NEXT
200 FOR I=0 TO 127
210 H(I)=ASC(MID$(A$,I+1,1))
220 NEXT
230 FOR I=0 TO 127
240 H(I+128)=ASC(MID$(B$,I+1,1))
250 NEXT
260 FIELD 1,32 AS F$(1),32 AS F$(2),32 AS F$(3),32 AS
F$(4),32 AS F$(5),32 AS F$(6),32 AS F$(7),32 AS F$(8)
270 H1(31)=H(31)
280 FOR I=3 TO 15
290 GET 1,I
300 FOR J=1 TO 8

```

```
310 F$=MID$(F$(J),6,11)
320 IF ASC(F$(J)) AND &H10 THEN GOSUB 510
330 NEXT
340 NEXT
350 FOR I=0 TO 255
360 IF H1(I)<>H(I) THEN PRINT"HIT-Fehler";I
370 NEXT
380 FIELD 1,128 AS A$,128 AS B$
390 AA$="" : BB$=""
400 FOR I=0 TO 127
410 AA$=AA$+CHR$(H1(I))
420 NEXT
430 FOR I=128 TO 255
440 BB$=BB$+CHR$(H1(I))
450 NEXT
460 LSET A$=AA$
470 LSET B$=BB$
480 POKE 23708,129
490 PUT 1,2
500 END
510 IF ASC(F$(J)) AND &H80 THEN 570
520 GOSUB 90
530 K=(J-1)*32+I-3
540 H1(K)=HC
550 PRINTF$,K,HC
560 RETURN
570 K=ASC(MID$(F$(J),2)):HC=H1(K):GOTO 530
580 IF ERR=46 THEN RESUME NEXT
590 ON ERROR GOTO 0
```

1.3.8 Datum und Uhrzeit

Datum und Uhrzeit werden automatisch per Interrupt aktualisiert. Sie können vom Basic aus mit TIME\$ abgefragt, aber nicht gesetzt werden.

In folgenden Speicherzellen sind Datum und Uhrzeit abgelegt:

4041H : Sekunde
 4042H : Minute
 4043H : Stunde
 4044H : Jahr
 4045H : Monat
 4046H : Tag

Mit POKE-Befehlen können Datum und Uhrzeit gesetzt werden. Das DOS stellt folgende Routinen zur Umwandlung von Datum und Uhrzeit in einen String zur Verfügung:

```
*****
*
* CONTIM : Zeit in String der Form hh:mm:ss umwandeln
*
*   E : CE6DH
*   R : AF, BC, DE, HL
*   I : HL = Zeiger auf Buffer (8 Bytes lang)
*   O : HL = Zeiger hinter ss
*       (HL-8), (HL-7) = hh
*       (HL-5), (HL-4) = mm
*       (HL-2), (HL-1) = ss
*
*****
```

Funktion:

Die Zeit aus den Speicherzellen 4041H-4043H wird im Format hh:mm:ss in den vom Benutzer definierten Buffer geschrieben.

Beispiel:

| | | |
|------|-----------------|------------------------|
| | ; Zeit ausgeben | |
| LD | HL,BUFFER | ; Bufferzeiger |
| PUSH | HL | ; retten |
| CALL | CONTIM | ; Zeit in Buffer |
| LD | (HL),0 | ; Bufferende markieren |
| POP | HL | ; Bufferanfang zurück |
| CALL | PRINTT | ; Buffer ausgeben |

```

*****
*
* CONDAT : Datum in String der Form tt.mm.jj umwandeln
*
*      E : CE70H
*      R : AF, BC, DE, HL
*      I : HL = Zeiger auf Buffer (8 Bytes lang)
*      O : HL = Zeiger hinter jj
*          (HL-8), (HL-7) = tt
*          (HL-5), (HL-4) = mm
*          (HL-2), (HL-1) = jj
*
*****

```

Funktion:

Das Datum aus den Speicherzellen 4044H-4046H wird im Format hh:mm:ss in den vom Benutzer definierten Buffer geschrieben.

Beispiel:

| | | |
|------|-----------------|------------------------|
| | ; Zeit ausgeben | |
| LD | HL,BUFFER | ; Bufferzeiger |
| PUSH | HL | ; retten |
| CALL | CONDAT | ; Datum in Buffer |
| LD | (HL),0 | ; Bufferende markieren |
| POP | HL | ; Bufferanfang zurück |
| CALL | PRINTT | ; Buffer ausgeben |

Folgendes Beispielprogramm überträgt Datum und Uhrzeit in einen Buffer und gibt diesen bei jeder Änderung der Sekunde auf dem Bildschirm aus.

```

100 ;*****
110 ;*
120 ;* BEISPIEL
130 ;*
140 ;* ZEIT ABFRAGEN UND AUSGEBEN
150 ;*
160 ;* (C) 1985 LUIDGER RÖCKRATH
170 ;*
180 ;*****
190 ;
200 CONTIM EQU 0CE6DH ; ZEIT IN BUFFER
210 CONDAT EQU 0CE70H ; DATUM IN BUFFER
220 PRINTT EQU 2B75H ; BUFFER AUSGEBEN
230 CURSOR EQU 4020H ; CURSORADRESSE
240 ;
250 START LD HL,4400H+40-18
260 LD (CURSOR),HL ; CURSOR SETZEN
270 LD HL,BUFFER ; DATUM
280 CALL CONDAT ; IN BUFFER
290 LD (HL),' ' ; SPACE DAHINTER
300 INC HL
310 LD A,(BUFFER+16) ; SECEINER RETTEN
320 PUSH AF
330 CALL CONTIM ; ZEIT IN BUFFER
340 LD (HL),13 ; CR IN BUFER
350 INC HL
360 LD (HL),0 ; BUFFERENDE MARKIEREN
370 POP BC ; SECEINER NACH B
380 LD A,(BUFFER+16) ; NEUER SECEINER
390 CP B ; AENDERUNG
400 JR Z,START ; NEIN -->
410 LD HL,BUFFER
420 CALL PRINTT ; BUFFER AUSGEBEN
430 JR START
440 ;
450 BUFFER DEFS 17
460 END START

```

Teil 2 : Hilfsprogramme Anleitungen und Quellprogramme

=====

Die Quellprogramme, die folgen, sind für den COLASM (Kassetten- oder Diskettenversion) ausgelegt. Die ORG-Adressen sind so festgelegt, daß die Programm im höchsten Speicher des 32K-Systems residieren. Da der COLASM keine Assemblerlistings mit Quell- und Maschinencode erstellen kann, wurden die Listings mit dem EDTASM unter GDOS mit einem Genie III s erstellt. Der Quellcode beginnend mit der Zeilennummer kann aber auch auf einem Colour-Genie mit COLASM verarbeitet werden, die Syntax des Quellcodes ist bei EDTASM und COLASM identisch. Alle Programme außer COLBAS (zu lang) wurden auch mit dem COLASM assembliert und erzeugten dort einen einwandfrei lauffähigen Objektcode. Mnemonics, die vom COLASM fehlerhaft assembliert werden (ADD HL,SP), wurden durch DEFB ersetzt.

Die Kassette bzw. Diskette mit allen Quellprogramme (inkl. der Beispiele aus Teil 1) ist natürlich direkt vom COLASM zu verarbeiten. Alle Quellprogramme sind dort als Basicprogramme aufgezeichnet, so wie es der COLASM fordert.

Wer über keinen Assembler verfügt (Der Assembler COLASM kann von uns für DM 69,-- (auf Kassette) bzw. 95,-- (auf Diskette) erworben werden), kann auch mit einem Monitor (z. B. CBUG) die Programme direkt in den Speicher schreiben. Wird dabei die Startadresse (ORG) geändert werden, muß bei allen Befehlen, die davon beeinflußt werden, die Adresse entsprechend geändert werden.

Auf der Quellprogramm-kassette bzw -diskette für DM 99,-- DM sind folgende Sourcen aufgezeichnet:

B1/ASM
B2/ASM
B3/ASM
B4/ASM
B5/ASM
B6/ASM
B7/ASM
B8/ASM
B9/ASM
VIDEO/ASM
SEDIT/ASM
COLBAS/ASM (auf Diskette nur Teile)
ARRAYD/ASM (nur Diskette)
ARRAY/ASM (nur Kassette)
COPBIT/ASM
SAVETAPE/ASM (nur Diskette)

Format der Diskette: 40 Spur, einseitig, doppelte Dichte

Auf der Kassette sind die genannten Quellprogramme in dieser Reihenfolge hintereinander aufgezeichnet.

2.2.1 Bildschirmeditor SEDIT

=====

Das kurze Maschinenprogramm SEDIT ermöglicht volles Bildschirmeditieren von Basicprogrammen, ähnlich wie bei den CBM-Rechnern. SEDIT vereinigt große Effizienz mit einfacher Bedienung, da komplizierte Manipulationen mit wenigen einfachen und selbstverständlichen Grundfunktionen durchgeführt werden können.

Funktion:

SEDIT wird durch Verändern des Displayvektors an den Basicinterpreter angebunden. Der Aufruf der OUTCH-Routine in der Hauptschleife zur Ausgabe des Promptzeichen kann so erkannt und unterdrückt werden. Stattdessen wird der Bildschirmeditor aufgerufen und beim Drücken von <ENTER> diejenige logische Zeile, in der sich der Cursor befindet, im I/O-Buffer an den Basicinterpreter übergeben.

Außerdem muß die OUTCH-Routine angezapft werden, um die Funktion des CR zu verändern. CR (=0DH) bewirkt normalerweise bei der Ausgabe per OUTCH das Positionieren des Cursors auf den Anfang der folgenden Zeile und das Löschen dieser Zeile. Unter SEDIT wird vor der Positionierung der Rest der Zeile, in der sich der Cursor befindet, gelöscht und das Ende der Zeile durch ein 80H (Grafikspace, unsichtbar) markiert. Die folgende Zeile, auf deren Anfang der Cursor positioniert wird, wird nicht verändert.

Durch die Markierung des Ende jeder logischen Zeile ist die Verarbeitung von beliebig langen Zeilen durch SEDIT möglich. Die Markierungen werden beim Scrollen automatisch mitgenommen und durch SEDIT beim Einfügen, Löschen, ... richtig modifiziert, wenn dies notwendig ist.

Der CBM-Bildschirmeditor arbeitet nicht mit solchen Markierung sondern mit Bitmasken, die angeben, ob aufeinanderfolgende Bildschirmzeilen zusammengehören. Da diese Bitmasken bei jedem Scroll verändert werden müssen, verbat sich dieses Vorgehen beim Colour-Genie, wenn eine völlige Neuprogrammierung der OUTCH-Routine vermieden werden soll. Außerdem ermöglicht das Konzept mit den Markierungen interessante Möglichkeiten zum Editieren, wie z. B. die Verknüpfung zweier Zeilen.

So arbeiten Sie mit SEDIT:

Sobald SEDIT geladen und initialisiert ist, erscheint der neue Cursor: ein blinkendes Quadrat. Dieser erscheint fortan immer, wenn im normalen Betrieb das Promptzeichen erscheinen würde, also nicht unter AUTO, SYSTEM, EDIT oder während der Programmausführung.

Sobald der blinkende Cursor zu sehen ist, eröffnet sich Ihnen eine Vielfalt neuer Möglichkeiten das Programm zu verändern. Da der Cursor mit den vier Pfeiltasten zerstörungsfrei über den gesamten Bildschirm bewegt werden kann, wird auf der Cursorposition abwechselnd das Cursorzeichen (weißes Quadrat) und das Zeichen, welches sich auf dieser Position befand, angezeigt.

Auf der Cursorposition kann der Bildschirminhalt nun einfach überschrieben werden, aber alle Änderungen werden erst in den Speicher übernommen, wenn die ENTER-Taste gedrückt wird (s.u.).

Mit den vier Pfeiltasten kann der Cursor bewegt werden. Stößt er bei Vor- oder Rückwärtsbewegung auf das Zeilenende oder den -anfang, geht er automatisch in die vorhergehende oder nachfolgende Zeile. Bei der Auf- und Abwärtsbewegung bleibt er am Bildschirmrand stehen.

(Da <^> zur Cursorbewegung dient, ergibt <SHIFT><^> fortan den Exponentiationspfeil.)

Bei den im folgenden dargestellten Funktionen werden von SEDIT keine physikalischen Zeilen (d.h. die 64 Zeichen einer Bildschirmzeile), sondern logische Zeilen (d.h. Zeilen, die zwar länger als 64 Zeichen sein können, aber sinngemäß zusammengehören.) verarbeitet.

Anmerkung zu den Beispielen:

_ : Cursorposition

\$: Taste festhalten bis Cursor auf gewünschter Position

<SHIFT><←> <DELETE> :

Das Zeichen auf der Cursorposition wird gelöscht; der Rest der logischen Zeile rückt nach.

Beispiel:

| | |
|-------------------------------|------------|
| 10 FOR I=0 TO 10:PRINT I:NEXT | <SHIFT><←> |
| 10 FOR I=0 TO 10:RINT I:NEXT | <SHIFT><←> |
| 10 FOR I=0 TO 10:INT I:NEXT | <SHIFT><←> |
| 10 FOR I=0 TO 10:NT I:NEXT | <SHIFT><←> |
| 10 FOR I=0 TO 10:T I:NEXT | <SHIFT><←> |
| 10 FOR I=0 TO 10:_I:NEXT | <SHIFT><←> |
| 10 FOR I=0 TO 10:I:NEXT | <SHIFT><←> |
| 10 FOR I=0 TO 10:;NEXT | <SHIFT><←> |
| 10 FOR I=0 TO 10:NEXT | |

(Anstatt 8 mal die <SHIFT> und <←>-Taste zu drücken, können Sie diese Tasten auch einfach festhalten, bis der 2. Doppelpunkt unter dem Cursor erscheint (s. u. Repeatfunktion).

<SHIFT><→> <INSERT> :

Nach einmaligem Drücken von <SHIFT><→> befindet sich SEDIT im Insert-Mode, der erst bei Drücken eines nichtdruckenden Zeichen (z.B. Cursorbewegung oder ENTER) wieder verlassen wird. Im Insert-Mode wird jedes druckende Zeichen an der Cursorposition eingefügt und der Rest der logischen Zeile wird weitergeschoben. Auch bei Überschreiten des Zeilenendes wird der Bildschirminhalt richtig korrigiert.

Beispiel:

| | |
|---------------------|------------|
| 10 A=MID\$(A\$,2) | <SHIFT><→> |
| 10 A=MID\$(A\$,2) | <,> |
| 10 A=MID\$(A\$,2,) | <1> |
| 10 A=MID\$(A\$,2,1) | <ENTER> |

(Durch das einmalige <SHIFT><→> wird SEDIT in den Insert-Modus versetzt. Alle druckenden (d.h. direkt auf dem Bildschirm darstellbaren Zeichen) werden jetzt an der Cursorposition eingefügt, und der Rest der Zeile wird verschoben. Durch Eingabe eines nichtdruckenden Zeichen (wie z.B.: <ENTER>) wird dieser Modus verlassen und im Beispiel die Zeile in Ihrer neuen Form abgespeichert.

<SHIFT><@> (SKIP) :

Vom Cursor an wird der Rest der logischen Zeile gelöscht.

Beispiel:

```
10 ? I,I*I,I*I*I
10 ? I,I*I_
```

<SHIFT><@>

<BREAK> :

Vom Bildschirmende wird die erste freie Zeile zur bequemen Befehlseingabe gesucht, und der Cursor dort positioniert.

<CLEAR> :

(Wie gehabt.)

<ENTER> :

Der Inhalt derjenigen logischen Zeile, in der sich der Cursor befindet, wird über den Tastaturtreiber an den Basicinterpreter übergeben (Ist die Zeile länger als 240 Zeichen, wird ein LS-Error erkannt.). Der Cursor befindet sich danach in der nächsten logischen Zeile.

Besondere Tricks

Mit SEDIT können Programmodifikationen, die sonst kompliziert und langwierig sind, einfach und komfortabel erledigt werden. Hier sind einige wichtige Tricks zusammengetragen:

1) Verschieben von Programmzeilen

Mit SEDIT können einfach Zeilennummern geändert und somit Programmzeilen verschoben oder dupliziert werden.

Beispiel:

```
10 ? I".ELEMENT"A(I)          20<ENTER>
20 ? I".ELEMENT"A(I)          LIST<ENTER>
...
```

2) Editieren nach Syntax-Fehlern

Bei Syntax-Fehlern springt der TRS-80 automatisch in den Zeileneditor. Unter SEDIT verläßt man diesen am besten sofort mit ENTER (was einen Ausdruck der Zeile bewirkt) und editiert sie sodann mit SEDIT.

3) Verknüpfen von Programmzeilen

Zwei Programmzeilen können einfach zu einer einzigen verbunden werden:

3.1) Überschreiben des letzten Zeichens der 1. logischen Zeile (letzte Bildschirmspalte). Der Cursor befindet sich sodann in der 2. Zeile und beide bilden nun eine logische Zeile. Mit Delete werden nun nur noch die überflüssigen Leerzeichen und die 2. Zeilennummer entfernt.

3.2) Löschen des letzten Zeichens (s.o.) mit Delete. Die 2. Zeile rückt dann nach. Weiter wie bei 3.1).

Beispiel:

| | Bildschirmrand | |
|-------------------------|----------------|--------------|
| 10 FOR I=0 TO 10 | 1 | <SHIFT><←> |
| 20 NEXT | 2 | \$<←> |
| 10 FOR I=0 TO 10 | 2 | \$<SHIFT><←> |
| 0 NEXT | | <SHIFT><←> |
| 10 FOR I=0 TO 10_ | | <SHIFT><←> |
| 0 NEXT | | <: > |
| 10 FOR I=0 TO 1020 NEXT | | <ENTER> |
| 10 FOR I=0 TO 100_ NEXT | | |
| 10 FOR I=0 TO 10_ NEXT | | |
| 10 FOR I=0 TO 10: NEXT | | |

4) Übernahme von Programmausdrucken

Mit SEDIT können auch Ausdrücke von Programmen einfach in den Speicher übernommen werden: z.B. können Maschinenprogramme von einem kleinen Programm mit Zeilennummer und DATA-Anweisung auf den Bildschirm gedruckt werden und dann mit SEDIT in den Speicher übernommen werden.

5) Wiederholung von ähnlichen Direktanweisungen

Direktanweisungen können ohne aufwendige Tipperei bequem mit verschiedenen Parametern durchlaufen werden.

Kleinbuchstaben und Umlaute

Unter SEDIT ist eine vernünftige Verwendung der Kleinbuchstaben wie auf einer Schreibmaschine möglich. Ohne <SHIFT> oder <LOCK> groß und mit klein. Will man den Originalmodus wieder erreichen, so kann man dies mit <CTRL> und <0>. Die Umlaute werden durch Drücken der Tasten <CTRL> und <A, O, U, a, o, u, s> erreicht (hintereinander). Sie werden auf dem Bildschirm richtig dargestellt und im Programmtext mit den üblichen Codes 5BH..5DH, 7BH..7EH abgelegt, so daß ohne weiteres auf einem Drucker Listings vom Programm gemacht werden können. Bei Eingaben im Programm können die Umlaute nicht genutzt werden.

Zusammenfassung

| | |
|------------|---|
| <→> | Cursor 1 Zeichen nach rechts |
| <←> | Cursor 1 Zeichen nach links |
| <↑> | Cursor 1 Zeile hoch |
| <↓> | Cursor 1 Zeile runter |
| <SHIFT><←> | DELETE: Zeichen auf Cursorposition löschen Rest der Zeile nach links rücken. |
| <SHIFT><→> | INSERT: Insert-Modus setzen. Folgende Zeichen an Cursorposition einfügen |
| <SHIFT><@> | SKIP: Rest der Zeile löschen |
| <BREAK> | CMD: Cursor in erste freie Zeile vom Bildschirmende her positionieren |
| <CLEAR> | CLEAR: Bildschirm löschen |
| <ENTER> | RETURN: Zeile an Basicinterpreter übergeben. |

Kleinbuchstaben und Umlaute:

| | |
|-----------|---|
| <CTRL><0> | Umschalten in Schreibmaschinenmodus und zurück |
| <CTRL><A> | A |
| <CTRL><a> | a |
| <CTRL><O> | Ö |
| <CTRL><o> | ö |
| <CTRL><U> | U |
| <CTRL><u> | ü |
| <CTRL><s> | ß |

```

00100      ; ** TRS-80-SCREEN-EDITOR **
00110      ; COPYRIGHT BY LUDGER ROEDERATH
00120      ; FAISERSTR. 54
00130      ; 4050 MÜNCHENGLADBACH 1
00140      ; JUNI 1981
00150      ; BASIC-UNTERPROGRAMME:
002B      00160 INCH EQU 002BH      ; TASTATURABFRAGE
0033      00170 OUTCH EQU 0033H     ; ZEICHENAUSGABE
318E      00180 SCROLL EQU 318EH    ; BILDSCHIRMSCROLL
01C9      00190 CLEAR EQU 01C9H     ; BILDSCHIRM LÖSCHEN
31BF      00200 CLEOL EQU 31BFH     ; BIS ZEILENENDE LÖSCHEN
3165      00210 CR EQU 3165H        ; CURSOR ZUM ZEILENANFANG
3203      00220 DOWN EQU 3203H      ; CURSOR EINE ZEILE RUNTER
1A7B      00230 RET EQU 1A7BH       ; RUECKSPRUNGADRESSE
19A2      00240 ERROR EQU 19A2H     ; BASIC-ERRORROUTINE
0060      00250 DELAY EQU 0060H     ; ZEITSCHLEIFE
00260      ; BASIC-ZEIGER:
4020      00270 CURSOR EQU 4020H    ; CURSORADRESSE
4016      00280 KEYVEC EQU 4016H    ; KEYBOARD-TREIBER-VEKTOR
401E      00290 DISVEC EQU 401EH    ; DISPLAY-TREIBER-VEKTOR
40A7      00300 BUFFER EQU 40A7H    ; I/O-BUFFER ANFANG
4036      00310 LAST EQU 4036H
F801      00320 FE7BD EQU 0F801H
00330      ; INTERNE SYMBOLE
0080      00340 MARKER EQU 80H      ; TRENNT LOGISCHE ZEILEN
0400      00350 EZEIT EQU 400H
0080      00360 RFREQ EQU 128
00370      ; INITIALISIERUNG:
00380      ; ASEG
BC50      00390 ORG 0BC50H
00400      ; SEDIT INITIALISIEREN
BC50 2A1E40 00410 INIT LD HL,(DISVEC)
BC53 2296BC 00420 LD (OUT+1),HL
BC56 2168BC 00430 LD HL,SEDI      ; DISPLAYVEKTOR ÄNDERN
BC59 221E40 00440 LD (DISVEC),HL
00450      ;
BC5C 2A1640 00460 LD HL,(KEYVEC)
BC5F 21D2BE 00470 LD HL,NEWKEY
BC62 221640 00480 LD (KEYVEC),HL
BC65 C3191A 00490 JP 1A19H
00500      ;
00510      ;
00520      ; HAUPTPROGRAMM
BC68 210E00 00530 SEDIT LD HL,14    ; RUECKSPRUNGADRESSE
BC6B 39      00540 DEFB 39H
00550      ; = ADD HL,SP      ; DURCH STACKMANIPULATION
00560      ; wird von COLASM falsch übersetzt
BC6C 5E      00570 LD E,(HL)      ; ERMITTELN
BC6D 23      00580 INC HL
BC6E 56      00590 LD D,(HL)
BC6F E5      00600 PUSH HL
BC70 217B1A 00610 LD HL,RET
BC73 DF      00620 RST 18H      ; =RET
BC74 E1      00630 POP HL
BC75 2845    00640 JR Z,EDIT      ; JA
BC77 79      00650 LD A,C      ; NORMALE AUSGABE
BC78 21E0BF 00660 LD HL,CTAB2
BC7B CD62BF 00670 CALL CONVER
BC7E 2011    00680 JR NZ,01
BC80 2A2040 00690 LD HL,(CURSOR)

```



```

BC83 F5      00700      PUSH AF
BC84 0E3A     00710      LD C,':'
BC86 A7       00720      AND A
BC87 CD95BC   00730      CALL OUT
BC8A F1       00740      POP AF
BC8B 2A2040   00750      LD HL,(CURSOR)
BC8E 2B       00760      DEC HL
BC8F 77       00770      LD (HL),A
BC90 C9       00780      RET
BC91 4F       00790 01    LD C,A
BC92 D60D     00800      SUB 0DH
BC94 A7       00810      AND A          ;CARRY LOESCHEN!
BC95 C20000   00820 OUT    JP NZ,0      ;NEIN
BC98 DD6E03   00830      LD      L,(IX+3)
BC9B DD6604   00840      LD      H,(IX+4)
BC9E DD7E05   00850      LD      A,(IX+5)
BCA1 A7       00860      AND      A
BCA2 2801     00870      JR      Z,000
BCA4 77       00880      LD      (HL),A
BCA5 E5       00890 000    PUSH    HL
BCA6 CD5F30   00900      CALL    305FH
BCA9 CD6531   00910      CALL    CR
BCAC 112800   00920      LD      DE,40
BCAF 19       00930      ADD     HL,DE
BCB0 EB       00940      EX      DE,HL
BCB1 1B       00950      DEC     DE
BCB2 E1       00960      POP     HL
BCB3 CD36BD   00970      CALL    SKIP1
BCB6 CDBABE   00980      CALL    CR1
BCB9 C30831   00990      JP      310BH
BCBC 23       01000 EDIT   INC     HL          ;NEUER STACK
BCBD F9       01010      LD      SP,HL
BCBE 3E0E     01020      LD      A,14          ;'-'-CURSOR OFF
BCC0 CD3300   01030      CALL    OUTCH
BCC3 D9       01040      EXX          ;FLAG LOESCHEN
BCC4 1E00     01050      LD      E,0
BCC6 D9       01060      EXX
BCC7 2A2040   01070 LOOP   LD      HL,(CURSOR)
BCCA CD2B00   01080 SCAN   CALL    INCH
BCCD A7       01090      AND     A          ;TASTE GEDRUECKT?
BCCE 2002     01100      JR      NZ,BRANCH      ;JA
BCD0 1BF8     01110      JR      SCAN          ;NEIN
BCD2 11C7BC   01120 BRANCH LD      DE,LOOP      ;RUECKSPRUNGADRESSE
BCD5 D5       01130      PUSH    DE
BCD6 FE60     01140      CP      'S'+20H      ;SKIP?
BCD8 2855     01150      JR      Z,SKIP      ;JA
BCDA FE20     01160      CP      ' '          ;ZEICHEN < ' '
BCDC 380A     01170      JR      C,CNTL      ;JA
BCDE D9       01180      EXX          ;INSERT-FLAG GESETZT
BCDF CB43     01190      BIT     0,E
BCE1 D9       01200      EXX
BCE2 C27FBD   01210      JP      NZ,INSERT      ;JA
BCE5 C33300   01220      JP      OUTCH      ;ZEICHEN AUSGEBEN
BCE8 D9       01230 CNTL   EXX          ;INSERT-FLAG SETZEN
BCE9 CBC3     01240      SET     0,E
BCEB D9       01250      EXX
BCEC FE19     01260      CP      19H          ;INSERT?
BCEE CB       01270      RET     Z          ;JA
BCEF F5       01280      PUSH    AF
BCF0 CD5F30   01290      CALL    305FH

```

```

BCF3 F1      01300      POP AF
BCF4 D9      01310      EXX          ;INSERT-FLAG ZURUECK SETZ.
BCF5 CB83    01320      RES 0,E
BCF7 D9      01330      EXX
BCF8 FE18    01340      CP 18H      ;DELETE?
BCFA 2862    01350      JR Z,DELETE ;JA
BCFC FE1F    01360      CP 1FH      ;CLEAR?
BCFE CAC901  01370      JP Z,CLEAR  ;JA
BD01 FE01    01380      CP 1        ;BREAK?
BD03 2838    01390      JR Z,BREAK  ;JA
BD05 FE0D    01400      CP 0DH      ;ENTER?
BD07 CA3DBE  01410      JP Z,RETURN ;JA
BD0A 11D8FF  01420      LD DE,-40   ;ZEILE RAUF
BD0D FE18    01430      CP 18H      ;UP?
BD0F 2812    01440      JR Z,MOVE   ;JA
BD11 11FFFF  01450      LD DE,-1   ;ZEICHEN ZURUECK
BD14 D608    01460      SUB 8       ;BACK?
BD16 2808    01470      JR Z,MOVE   ;JA
BD18 110100  01480      LD DE,1    ;ZEICHEN VOR
BD1B 3D      01490      DEC A       ;ADVANCE?
BD1C 2805    01500      JR Z,MOVE   ;JA
BD1E 112800  01510      LD DE,40   ;ZEILE RUNTER
BD21 3D      01520      DEC A       ;DOWN?
BD22 C0      01530      RET NZ      ;NEIN
BD23 19      01540 MOVE ADD HL,DE   ;ZUM CURSOR ADDIEREN
BD24 CDC8BE  01550      CALL VIDEND ;CURSOR NOCH IM VIDEORAM?
BD27 C8      01560      RET Z       ;NEIN, AENDERUNG INSN.
BD28 7C      01570      LD A,H
BD29 FE43    01580      CP 43H
BD2B C8      01590      RET Z       ;NEIN
BD2C C30831  01600      JP 3108H
BD2F CD9EBE  01610 SH IP CALL SEARCH ;LETZTES ZEICHEN SUCHEN
BD32 EB      01620      EX DE,HL    ;LETZTES ZEICHEN
BD33 DF      01630      PST 18H     ;HINTER CURSOR?
BD34 EB      01640      EX DE,HL
BD35 D8      01650      RET C       ;JA
BD36 3620    01660 SH IP1 LD (HL),'' ;ZEICHEN LOESCHEN
BD38 DF      01670      PST 18H     ;LETZTES ZEICHEN GEL.?
BD39 23      01680      INC HL
BD3A 20FA    01690      JR NZ,SH IP1 ;NEIN
BD3C C9      01700      RET
BD3D 21E747  01710 BREAK LD HL,47E7H ;LETZTES VIDEORAM-BYTE
BD40 7E      01720 BREAK1 LD A,(HL)
BD41 FE20    01730      CP ''       ;BLANK?
BD43 2006    01740      JR NZ,BREAK2 ;NEIN
BD45 2B      01750      DEC HL      ;JA, WEITER ZURUECK
BD46 7C      01760      LD A,H      ;VIDEORAM-ANF. ERREICHT?
BD47 FE43    01770      CP 43H
BD49 20F5    01780      JR NZ,BREAK1 ;NEIN
BD4B FE80    01790 BREAK2 CP 80H
BD4D 2801    01800      JR Z,BREAK3
BD4F 23      01810      INC HL
BD50 CDC8BE  01820 BREAK3 CALL VIDEND
BD53 CC8E31  01830      CALL Z,SCROLL
BD56 222040  01840      LD (CURSOR),HL ;NEUER CURSOR
BD59 7E0D    01850      LD A,0DH   ;CR
BD5B C33300  01860      JP OUTCH
BD5E 23      01870 DELETE INC HL   ;LETZTES ZEICHEN AB
BD5F CD9EBE  01880      CALL SEARCH ;CURSOR+1 SUCHEN
BD62 2B      01890      DEC HL

```

| | | | |
|-------------|-------|---------------------|-----------------------------|
| BD63 DF | 01900 | RST 18H | ; MIT CURSOR VERGL. |
| BD64 2816 | 01910 | JR Z, DEL | ; SPRUNG, WENN GLEICH |
| BD66 D0 | 01920 | RET NC | ; FERTIG WENN HINTER CURSOR |
| BD67 54 | 01930 | LD D, H | ; CURSOR=ZIELADRESSE |
| BD68 5D | 01940 | LD E, L | ; FUER BLOCKMOVE |
| BD69 23 | 01950 | INC HL | ; CURSOR+1=QUELLADR. |
| BD6A D5 | 01960 | PUSH DE | |
| BD6B C5 | 01970 | PUSH BC | |
| BD6C EDB0 | 01980 | LDIR | ; MOVE |
| BD6E C1 | 01990 | POP BC | |
| BD6F D1 | 02000 | POP DE | |
| BD70 E5 | 02010 | PUSH HL | |
| BD71 2100AC | 02020 | LD HL, 0F000H-4400H | |
| BD74 19 | 02030 | ADD HL, DE | |
| BD75 54 | 02040 | LD D, H | |
| BD76 5D | 02050 | LD E, L | |
| BD77 23 | 02060 | INC HL | |
| BD78 EDB0 | 02070 | LDIR | |
| BD7A E1 | 02080 | POP HL | |
| BD7B 2B | 02090 | DEC HL | ; ZURUECK ZUM LETZTEN BYTE |
| BD7C 3620 | 02100 | LD (HL), ' ' | ; MIT SPACE UEBERSCHR. |
| BD7E C9 | 02110 | RET | |
| BD7F F5 | 02120 | PUSH AF | ; ZEICHEN RETTEN |
| BD80 CD9EBE | 02130 | CALL SEARCH | ; LETZTES ZEICHEN SUCHEN |
| BD83 7E | 02140 | LD A, (HL) | ; CURSOR AUF MARKER? |
| BD84 FE80 | 02150 | CP MARKER | |
| BD86 2009 | 02160 | JR NZ, L1 | ; NEIN |
| BD88 D9 | 02170 | EXX | ; FLAG SETZEN |
| BD89 CBCB | 02180 | SET 1, E | |
| BD8B D9 | 02190 | EXX | |
| BD8C 54 | 02200 | LD D, H | ; ZEIGER AUF LETZTES Z. |
| BD8D 5D | 02210 | LD E, L | ; =CURSOR-1 |
| BD8E 1B | 02220 | DEC DE | |
| BD8F 1806 | 02230 | JR L2 | |
| BD91 DF | 02240 | L1 RST 18H | ; VERGL. CURSOR-LETZTES Z. |
| BD92 2803 | 02250 | JR Z, L2 | ; SPRUNG, WENN GLEICH |
| BD94 D239BE | 02260 | JF NC, L3 | ; CURSOR HINTER L.Z. |
| BD97 7B | 02270 | L2 LD A, E | ; L.Z. AM ZEILENENDE? |
| BD98 E5 | 02280 | PUSH HL | |
| BD99 D5 | 02290 | PUSH DE | |
| BD9A EB | 02300 | EX DE, HL | |
| BD9B 11A3BD | 02310 | LD DE, L111 | |
| BD9E D5 | 02320 | PUSH DE | |
| BD9F E5 | 02330 | PUSH HL | |
| BDA0 C34C03 | 02340 | JF 34CH | |
| BDA3 D1 | 02350 | L111 POP DE | |
| BDA4 E1 | 02360 | POP HL | |
| BDA5 FE26 | 02370 | CP 26H | |
| BDA7 2072 | 02380 | JR NZ, L4 | ; NEIN |
| BDA9 E5 | 02390 | PUSH HL | ; ENDE DER LETZTEN ZEILE |
| BDAA 21E647 | 02400 | LD HL, 47E6H | |
| BDAD DF | 02410 | RST 18H | |
| BDAE E1 | 02420 | POP HL | |
| BDAF 2019 | 02430 | JR NZ, L5 | ; NEIN |
| BDB1 D5 | 02440 | PUSH DE | ; JA, SCROLL |
| BDB2 11DBFF | 02450 | LD DE, -40 | |
| BDB5 19 | 02460 | ADD HL, DE | |
| BDB6 222040 | 02470 | LD (CURSOR), HL | |
| BDB9 D9 | 02480 | EXX | |
| BDBA D5 | 02490 | PUSH DE | |

```

BDBB D9      02500      EXX
BDBC C5      02510      PUSH BC
BDBD CD8E31  02520      CALL SCROLL
BDC0 C1      02530      POP BC
BDC1 D9      02540      EXX
BDC2 D1      02550      POP DE
BDC3 D9      02560      EXX
BDC4 D1      02570      POP DE
BDC5 21D8FF  02580      LD HL,OFFD8H      ;DE=DE-40
BDC8 19      02590      ADD HL,DE
BDC9 EB      02600      EX DE,HL
BDCA 21BE47  02610 L5    LD HL,47BEH      ;MUSS EIN REST NACH
BDCD DF      02620      RST 19H      ;UNTEN VERSCHOBEN WERDEN
BDCE D5      02630      PUSH DE
BDCF 23      02640      INC HL
BDD0 23      02650      INC HL
BDD1 2830    02660      JR Z,L7      ;NEIN
BDD3 13      02670      INC DE      ;NAECHSTES ZEICHEN MARKIER?
BDD4 1A      02680      LD A,(DE)
BDD5 FE80    02690      CP MARKER
BDD7 202A    02700      JR NZ,L7      ;NEIN, KEINE VERSCHIEBUNG
BDD9 EB      02710      EX DE,HL      ;HL=ANFANG DER NAECHSTEN Z.
BDDA 23      02720      INC HL
Bddb 1B      02730      DEC DE      ;QUELLADRESSE (7FBF)
BDDC DDE5    02740      PUSH IX
BDD E DD21E747 02750      LD IX,47E7H      ;ZIELADRESSE
BDE2 1A      02760 L6    LD A,(DE)
BDE3 DD7700  02770      LD (IX+0),A
BDE6 E5      02780      PUSH HL
BDE7 2100AC  02790      LD HL,0F000H-4400H
BDEA 19      02800      ADD HL,DE
BDEB 7E      02810      LD A,(HL)
BDEC E1      02820      POP HL
BDED D5      02830      PUSH DE
BDEE DDE5    02840      PUSH IX
BDF0 1100AC  02850      LD DE,0F000H-4400H
BDF3 DD19    02860      ADD IX,DE
BDF5 DD7700  02870      LD (IX+0),A
BDF8 DDE1    02880      POP IX
Bdfa D1      02890      POP DE
BDFB DF      02900      RST 18H      ;FERTIG?
BDFC 1B      02910      DEC DE
BDFD DD2B    02920      DEC IX
Bdff 20E1    02930      JR NZ,L6      ;NEIN
BE01 DDE1    02940      POP IX
BE03 C5      02950 L7    PUSH BC
BE04 E5      02960      PUSH HL
BE05 1128AC  02970      LD DE,0F000H-4400H+40
BE08 19      02980      ADD HL,DE
BE09 7E      02990      LD A,(HL)
BE0A E3      03000      EX (SP),HL
BE0B F5      03010      PUSH AF
BE0C CDBF31  03020      CALL CLEOL      ;FREIGEWORDENE ZEILE
BE0F F1      03030      POP AF
BE10 E3      03040      EX (SP),HL
BE11 77      03050      LD (HL),A
BE12 E1      03060      POP HL
BE13 C1      03070      POP BC
BE14 112700  03080      LD DE,39
BE17 19      03090      ADD HL,DE

```

```

BE18 D1      03100      POP DE
BE19 3680    03110      LD (HL),MARKER
BE1B D9      03120 L4    EXX          ;FLAG PRUEFEN
BE1C CB4B    03130      BIT 1,E
BE1E CB8B    03140      RES 1,E      ;UND LOESCHEN
BE20 D9      03150      EXX
BE21 2016    03160      JR NZ,L3      ;GESETZT?--KEIN MOVE
BE23 62      03170      LD H,D        ;EIN ZEICHEN VORWAERTS
BE24 6B      03180      LD L,E        ;VERSCHIEBEN
BE25 13      03190      INC DE
BE26 E5      03200      PUSH HL
BE27 D5      03210      PUSH DE
BE28 C5      03220      PUSH BC
BE29 EDB8    03230      LDDR
BE2B C1      03240      POP BC
BE2C D1      03250      POP DE
BE2D 2100AC  03260      LD HL,0F000H-4400H
BE30 19      03270      ADD HL,DE
BE31 E3      03280      EX (SP),HL
BE32 1100AC  03290      LD DE,0F000H-4400H
BE35 19      03300      ADD HL,DE
BE36 D1      03310      POP DE
BE37 EDB8    03320      LDDR
BE39 F1      03330 L3    POP AF      ;NEUES ZEICHEN EINFUEGEN
BE3A C33300  03340      JP OUTH
BE3D CD9EBE  03350 RETURN CALL SEARCH ;LETZTES ZEICHEN SUCHEN
BE40 E5      03360      PUSH HL      ;CURSOR RETTEN
BE41 1805    03370      JR R6
BE43 7E      03380 R1    LD A,(HL)   ;ERSTES ZEICHEN SUCHEN
BE44 FE80    03390      CP MARKER
BE46 2806    03400      JR Z,R4
BE48 2B      03410 R6    DEC HL
BE49 7C      03420      LD A,H
BE4A FE43    03430      CP 43H      ;ANF. DES VIDEORAM?
BE4C 20F5    03440      JR NZ,R1     ;NEIN
BE4E 23      03450 R4    INC HL
BE4F 7E      03460      LD A,(HL)
BE50 FE20    03470      CP ' '
BE52 28FA    03480      JR Z,R4
BE54 0600    03490      LD B,0      ;ZAEHLER
BE56 DD2AA740 03500      LD IX,(BUFFER) ;IN I/O-BUFFER
BE5A 7E      03510 R2    LD A,(HL)   ;UEBERTRAGEN
BE5B E5      03520      PUSH HL
BE5C 21E8BF  03530      LD HL,CTAB3
BE5F C5      03540      PUSH BC
BE60 CD62BF  03550      CALL CONVER
BE63 C1      03560      POP BC
BE64 E1      03570      POP HL
BE65 DD7700  03580      LD (IX+0),A
BE68 DF      03590      RST 16H      ;FERTIG?
BE69 23      03600      INC HL
BE6A DD23    03610      INC IX
BE6C 280D    03620      JR Z,R3      ;JA
BE6E 3007    03630      JR NC,LEER   ;LEERE ZEILE
BE70 10E8    03640      DJNZ R2      ;256 ZEICHEN UEBERTR.
BE72 1E1C    03650      LD E,1CH    ;LS-ERROR
BE74 C3A219  03660      JP ERROR
BE77 DD2B    03670 LEER  DEC IX      ;ZAEHLER AUF ANFANG
BE79 E1      03680      POP HL      ;CURSOR ZURUECK
BE7A E5      03690      PUSH HL

```

```

BE7B D1      03700 R3      POP DE
BE7C DD360000 03710      LD (IX+0),0      ;ENDE MARKIEREN
BE80 CD6531   03720      CALL CP          ;IN NAECHSTE ZEILE
BE87 112800   03730      LD DE,40
BE86 19       03740      ADD HL,DE
BE87 CD8ABE   03750      CALL CP1
BE8A CD0831   03760      CALL 3108H
BE8D 2AA740   03770      LD HL,(BUFFER) ;PARAMETER FUER BASIC
BE90 2B       03780      DEC HL
BE91 3E0F     03790      LD A,15
BE93 CD3300   03800      CALL OUTCH
BE96 AF       03810      XOR A
BE97 329940   03820      LD (4099H),A
BE9A D1       03830      POP DE          ;STACK KORRIG.
BE9B C37E1A   03840      JP RET+3       ;ZURUECK ZUM BASIC
BE9E E5       03850 SEARCH PUSH HL      ;CURSOR RETTEN
BE9F 010000   03860      LD BC,0       ;ZEICHENZAehler
BEA2 7E       03870 S1      LD A,(HL)   ;MARKER SUCHEN
BEA3 FE80     03880      CP MARKER
BEA5 2808     03890      JP C,S2
BEA7 23       03900      INC HL
BEA8 07       03910      INC BC
BEA9 CD08BE   03920      CALL VIDEND    ;BILDSCHIRMENDE?
BEAC 20F4     03930      JR NZ,S1      ;NEIN
BEAE 2B       03940      DEC HL        ;AUF LETZTEN BILDSCHIRMPL.
BEAF 03       03950 S2      INC BC
BEB0 2B       03960 S3      DEC HL        ;LEERZEICHEN IGNORIEREN
BEB1 0B       03970      DEC BC
BEB2 7E       03980      LD A,(HL)
BEB3 FE20     03990      CP ' '
BEB5 28F9     04000      JP Z,S3
BEB7 E3       04010      EX (SP),HL    ;CURSOR ZURUECK
BEB8 D1       04020      POP DE        ;DE=LETZTES ZEICHEN
BEB9 C9       04030      RET
BEBA CD08BE   04040 CR1     CALL VIDEND
BEBD CC8E31   04050      CALL 2,SCROLL
BEC0 222040   04060      LD (CURSOR),HL
BEC3 2B       04070      DEC HL        ;ZEILENENDE MARKIEREN
BEC4 3680     04080      LD (HL),MARKER
BEC6 23       04090      INC HL
BEC7 C9       04100      RET
BEC8 D5       04110 VIDEND PUSH DE
BEC9 11E847   04120      LD DE,47E8H
BECC DF       04130      RET 18H
BEDC D1       04140      POP DE
BECE DB       04150      RET C
BECF AF       04160      XOR A
BED0 7C       04170      LD A,H
BED1 C9       04180      RET
BED2 3AF9BF   04190 ;
BED3 3AF9BF   04200 NEWKEY LD A,(KEYZL)
BED5 A7       04210      AND A
BED6 2811     04220      JR Z,NF1
BED8 3D       04230      DEC A
BED9 32F9BF   04240      LD (KEYZL),A
BEDC 2AFABF   04250      LD HL,(KEYZG)
BEDF 7E       04260      LD A,(HL)
BEE0 23       04270      INC HL
BEE1 22FABF   04280      LD (KEYZG),HL
BEE4 A7       04290      AND A

```

```

BEE5 C0      04300      RET NZ
BEE6 3E0D    04310      LD A,0DH
BEE8 C9      04320      RET
BEE9 CD6EBF  04330 NZ 1  CALL REPEAT
BEEC CDF5BE  04340      CALL NZ 11
BEEF A7      04350      AND A
BEF0 C8      04360      RET Z
BEF1 32FEEF  04370      LD (ZEICHN),A
BEF4 C9      04380      RET
              04390      ;
BEF5 A7      04400 NZ 11  AND A
BEF6 C8      04410      RET Z
BEF7 FE41    04420      CP 'A'
BEF9 3814    04430      JR C,NZ 3
BEFB FE5B    04440      CP 'Z'+1
BEFD 3822    04450      JR C,LETTER
BEFF 280E    04460      JR Z,NZ 3
BF01 FE60    04470      CP 60H
BF03 3822    04480      JR C,FEY1
BF05 2808    04490      JR Z,NZ 3
BF07 FE78    04500      CP 'z'+1
BF09 3816    04510      JR C,LETTER
BF0B FE80    04520      CP 128
BF0D 381A    04530      JR C,FEY2
BF0F 211840  04540 NZ 3  LD HL,4018H
BF12 CB7E    04550      BIT 7,(HL)
BF14 2028    04560      JR NZ,NCTRL
BF16 4F      04570      LD C,A          ;1BH UND 5BH VERTAUSCHEN
BF17 E6BF    04580      AND 0BFH      ; (SHIFT A UND A)
BF19 FE1B    04590      CP 1BH
BF1B 79      04600      LD A,C
BF1D 2002    04610      JR NZ,E1
BF1E EE40    04620      XOR 40H
BF20 C9      04630 E1    RET
              04640      ;
BF21 21F8BF  04650 LETTER LD HL,SL
BF24 AE      04660      XOR (HL)
BF25 18E8    04670      JR NZ 3
              04680      ;
BF27 C61C    04690 FEY1  ADD A,7CH-4-5CH
BF29 D677    04700 FEY2  SUB 77H
BF2B 21494D  04710      LD HL,4349H
BF2E 110700  04720      LD DE,7
BF31 19      04730 FL2   ADD HL,DE
BF32 3D      04740      DEC A
BF33 20FC    04750      JR NZ,FL2
BF35 22FABF  04760      LD (FEYZ6),HL
BF38 7B      04770      LD A,E
BF39 32F9BF  04780      LD (FEYZ7L),A
BF3C 1812    04790      JR NKC1
              04800      ;
BF3E CBBE    04810 NCTRL RES 7,(HL)
BF40 FE30    04820      CP '0'
BF42 2811    04830      JR Z,NKC2
BF44 FE31    04840      CP '1'
BF46 DB      04850      RET C
BF47 FE39    04860      CP '9'
BF49 3014    04870      JR NC,NKC3
BF4B D631    04880      SUB '1'
BF4D CD2136  04890      CALL 3621H

```

```

BF50 AF      04900 NHC1      XOR A
BF51 32FEBF  04910          LD (ZEICHN),A
BF54 C9      04920          RET
BF55 3AF8BF  04930 NHC2      LD A,(SL)
BF58 EE20    04940          XOR 20H
BF5A 32F8BF  04950          LD (SL),A
BF5D 18F1    04960          JR NHC1
BF5F 21D8BF  04970 NHC3      LD HL,CTAB1
                        04980 ;
BF62 010800  04990 CONVER    LD BC,TABLEN
BF65 C5      05000          PUSH BC
BF66 EDB1    05010          CPIR
BF68 C1      05020          POP BC
BF69 C0      05030          RET NZ
BF6A 09      05040          ADD HL,BC
BF6B 2B      05050          DEC HL
BF6C 7E      05060          LD A,(HL)
BF6D C9      05070          RET
                        05080 ;
BF6E 3A80F8  05090 REPEAT    LD A,(0F880H)
BF71 FE12    05100          CP 12H
BF73 200D    05110          JP NZ,REP11
BF75 CDA938  05120          CALL 3DA9H
BF78 3A40F8  05130 REP12     LD A,(0F840H)
BF7B CB57    05140          BIT 2,A
BF7D 28F9    05150          JP Z,REP12
BF7F CDB038  05160          CALL 38B0H
BF82 213640  05170 REP11     LD HL,LAST      ;LETZTEN TASTATURSTATUS
BF85 0101F8  05180          LD BC,KEYBD      ;(4076-407C) MIT ZUSTAND
BF88 AF      05190          XOR A
BF89 57      05200          LD D,A          ;DER TASTATURVERGLEICHEN
BF8A 32FFBF  05210          LD (TM),A
BF8D 0A      05220 REP1      LD A,(BC)
BF8E 5F      05230          LD E,A
BF8F AE      05240          XOR (HL)      ;AENDERUNG
BF90 73      05250          LD (HL),E
BF91 202B    05260          JR NZ,REP2      ;JA
BF97 3AFFBF  05270          LD A,(TM)
BF96 B3      05280          OR E
BF97 32FFBF  05290          LD (TM),A
BF9A 14      05300          INC D
BF9B 2C      05310          INC L
BF9C CB01    05320          RLC C          ;ALLE ZEILEN VERGL.
BF9E 30ED    05330          JR NC,REP1      ;JA
BFA0 3AF0BF  05340          LD HL,(ZAEHLR)
BFA7 2B      05350          DEC HL
BFA4 7C      05360          LD A,H          ;=0?
BFA5 B5      05370          OR L
BFA6 3E00    05380          LD A,0
BFA8 200B    05390          JR NZ,RETREP
BFAA 3AFFBF  05400          LD A,(TM)
BFAD A7      05410          AND A
BFAE 2805    05420          JR Z,RETREP
BFBO 2E80    05430          LD L,RFREQU      ;ZAEHLER NEU SETZEN
BFBC 3AFEBF  05440          LD A,(ZEICHN)    ;ZEICHEN IN AUFU
BFBD 22FCBF  05450 RETREP    LD (ZAEHLR),HL
BFBB A7      05460          AND A
BFB9 CA0F04  05470          JP Z,40FH
BFBC E1      05480          POP HL
BFBD C9      05490          RET

```



```

05500 ;
BFBE 5F 05510 REP2 LD E,A ;ENTPRELLUNG:
BFBF C5 05520 PUSH BC ;19 MS WARTEN
BFC0 010005 05530 LD BC,500H
BFC3 CD6000 05540 CALL DELAY
BFC6 C1 05550 POP BC
BFC7 0A 05560 LD A,(BC) ;TASTE NOCH IMMER GEDR.
BFC8 A3 05570 AND E
BFC9 2803 05580 JR Z,REP3 ;NEIN
BFCE CD1A04 05590 CALL 41AH
BFCE 210004 05600 REP3 LD HL,EZEIT ;ZAEHLER MIT EINSATZZEIT
BFD1 22FCBF 05610 LD (ZAEHLR),HL
BFD4 32FEBF 05620 LD (ZEICHN),A
BFD7 C9 05630 RET
05640 ;
BFD8 41 05650 CTAB1 DEFM 'A0U5aous'
BFEO 5B 05660 CTAB2 DEFM 'A0U5a0U5'
BFEP 01 05670 CTAB3 DEFB 1
BFEP 02 05680 DEFB 2
BFEP 03 05690 DEFB 3
BFEB 08 05700 DEFB 8
BFEC 05 05710 DEFB 5
BFED 06 05720 DEFB 6
BFEE 07 05730 DEFB 7
BFEP 08 05740 DEFB 8
BFF0 5B 05750 CTAB4 DEFM 'A0U5a0U5'
0008 05760 TABLEN EQU 8 ; CTAB2-CTAB1
05770 ;
BFF8 20 05780 SL DEFB 20H
BFF9 00 05790 FKEYZL DEFB 0
BFFA 0000 05800 FKEYZG DEFW 0
BFFC 0000 05810 ZAEHLR DEFW 0
BFFE 00 05820 ZEICHN DEFB 0
BFFF 00 05830 TM DEFB 0
05840 ;
BC50 05850 END INIT
00000 TOTAL ERRORS
24324 TEXT AREA BYTES LEFT

```

2.2.1 Basicerweiterung COLBAS

=====

COLBAS bietet dem Cassettenbenutzer viele der Möglichkeiten, die sonst erst dem Diskettenbenutzer offenstehen.

Es ist folgendes unter COLBAS möglich:

- Flexibles Zusammenladen verschiedener Basicprogramme von Cassette
- Suchen von Zeichenfolgen im Basicprogrammtext
- USR0 bis USR9
- DEFUSR0 bis DEFUSR9
- DEFFN
- LINE INPUT
- MID\$ als Anweisung
- INSTR
- RESTORE n
- Formatiertes Listen von Programmen
- Entfernen unnötiger Leerzeichen und REM's

Funktion:

Die meisten Erweiterungen von NEUBAS werden über die unbenutzten Diskbasicbefehlsworte aufgerufen. Diese Anbindung ist ähnlich wie beim Programm PACK beschrieben programmiert, aber hier noch einfacher. Beim Programm PACK mußte in der Diskettenversion der Befehl KILL zum Löschen von Files erhalten bleiben. Daher wird nach Auftreten des KILL zunächst das Argument geprüft. Wenn es eine Zeichenkette ist, wird ein File gelöscht, andernfalls der Programmtext gepackt.

Der RESTORE N-Befehl wird durch Veränderung des RST 10H-Vektors implementiert. Dazu wird der RST 10H-Aufruf in der Interpretationsschleife, kurz vor dem Anspringen der einzelnen Routinen für die verschiedenen Anweisungen abgefangen.

MERGE und RSET (ZUSAMMENLADEN MEHRERER BASICPROGRAMME)

Mit MERGE und RSET wird es möglich mehrere schon auf Cassette vorhandene Basicprogramme zusammen in den Rechner zu bringen. Dies erweist sich als besonders wertvoll, wenn Sie in neu zu schreibende Programme bereits vorhandene Programmsegmente (insbesondere Unterprogramme aus einer schon vorhandenen Unterprogrammsammlung) einfügen möchten.

Durch Eingabe von 'MERGE' wird das momentan im Rechner befindliche Basicprogramm "unsichtbar" gemacht. Der Speicher scheint leer zu sein, wovon Sie sich durch Eingabe von 'LIST' überzeugen können. Sie können nun das zweite Programm wie gewohnt mittels 'CLOAD' laden. Danach steht Ihnen dieses zweite Programm so im Rechner zur Verfügung, wie Sie es nach normalen Ladevorgängen gewohnt sind. Das heißt, Sie können das Programm laufen lassen, listen ...

Um nun das "unsichtbare" Programm zusammen mit dem zugeladen zweiten Programm zum Vorschein zu bringen, müssen Sie dafür sorgen, daß die niedrigste Zeilennummer des zweiten Programmes größer ist als die höchste Zeilennummer des ersten (momentan "unsichtbaren") Programmes. Falls dies nicht ohnehin der Fall ist, kann mit Hilfe des 'NAME' Befehls leicht dafür gesorgt werden, daß das zweite Programm die erforderlichen höheren Zeilennummern erhält. (Bei dieser Umnummerierung ist es auch möglich, ein zugeladenes Unterprogramm nachträglich an das aufrufende Programm anzupassen.)

Durch Eingabe von 'RSET' wird nun das erste Programm neben dem Zweiten wieder "sichtbar" gemacht. Es stehen nun beide Programme zusammen im Speicher.

Nach diesem Verfahren können Sie nun noch beliebig viele weitere Programme zuladen.

Solange sich ein "unsichtbares" Programm im Rechner befindet erscheint zusätzlich zu jeder 'NEWBAS READY' Meldung der Text '* MERGE MODE *' auf dem Bildschirm. Dies soll Sie an das noch vorhandene, "unsichtbare" Programm erinnern.

GET (SUCHEN VON ZEICHENFOLGEN IM BASICPROGRAMMTEXT)

Mit GET können beliebige Zeichenfolgen im Basicprogrammtext gesucht werden. Dies ist nützlich wenn Sie z.B. feststellen wollen ob, bzw. wo bestimmte Variablennamen im Basicprogrammtext benutzt werden. Oder Sie möchten in einem längeren Basicprogrammtext eine bestimmte Stelle, deren Zeilennummer Sie nicht kennen, schnell finden.

Sie geben 'GET' ein, und der Rechner meldet sich mit 'STRING='. Nun können Sie die Zeichenfolge, nach welcher gesucht werden soll, eingeben. Hierbei müssen zwei Fälle unterschieden werden:

- 1) Sie suchen eine Variable (z.B. AS\$, N, N%) oder einen Ausdruck mit Basicschlüsselworten (z.B. GOTO 200, RETURN). In diesem Fall geben Sie die gesuchte Zeichenfolge ein, und schließen mit 'ENTER' ab.
- 2) Sie suchen Zeichenfolgen, die sich innerhalb von Anführungsstrichen ("), in DATA-Anweisungen oder in Bemerkungen (hinter REM bzw. ') befinden. Hier geben Sie vor der eigentlichen Zeichenfolge '@@' ein und schließen dann mit 'ENTER' ab.
(Wenn Sie also z.B. die Zeichenfolge "COPYRIGHT BY" suchen, geben sie '@@COPYRIGHT BY' ein.)

In beiden Fällen wird der Rechner darauf hin die Zeilennummern, in welchen die gesuchte Zeichenfolge auftritt, ausgeben. Sollte die Zeichenfolge nicht im Basicprogrammtext vorhanden sein, so wird dies anstelle der Zeilennummern gemeldet.

Bemerkung: Tritt die gesuchte Zeichenfolge in einer Zeile mehrmals auf, so wird auch dieselbe Zeilennummer entsprechend oft ausgegeben.
Dabei ist zu beachten, daß z.B. in der Zeichenfolge "****" die Zeichenfolge "*" dreimal vorkommt.

Bei 1) wird aus dem Suchtext vor dem Vergleich mit dem Basicprogramm der Zwischencode gebildet.
Bei 2) wird der Suchtext nicht verändert.

USR0 bis USR9 (AUFRUF VON 10 VERSCHIEDENEN MASCHINENPROGRAMMEN)

War es bislang für Sie nur möglich ein Maschinenprogramm aus einem Basicprogramm heraus aufzurufen, so können unter NEUBAS bis zu 10 verschiedene Maschinenprogramme aufgerufen werden.

Die Arbeitsweise der USR0 bis USR9 Funktionen ist genauso, wie in Ihrem Basichandbuch unter USR erläutert.

So rufen Sie z.B. mit

A=USR7(0)

das 7. Maschinenprogramm auf.

DEFUSR0 bis DEFUSR9 (STARTADRESSEN VON MASCHINENPROGRAMMEN)

Bislang mußte die Startadresse des Maschinenprogramms, welches mittels USR aufgerufen werden sollte, in zwei spezielle Speicherzellen "gePOKEt" werden.

Diese unpraktische Vorgehensweise wird unter NEUBAS durch die Befehle DEFUSR0 bis DEFUSR9 ersetzt.

Sie geben z.B.

DEFUSR7=15360 (oder hexadezimal DEFUSR7=&3C00)

ein und weisen damit dem 7. Maschinenprogramm die Startadresse 15360 (hex 3C00) zu.

DEFFN (VOM ANWENDER DEFINIERBARE FUNKTIONEN)

Mit DEFFN können Sie Funktionen, welche häufig in einem Programm benutzt werden, einmal definieren und dann im Laufe des Programms beliebig oft durch den Funktionsnamen aufrufen.

DEFFN hat folgendes Format:

DEFFN FKTN (VARLIST) = AUSDRUCK

FKTN ist der Name der Funktion, die hier definiert wird. Für diesen Namen gelten die gleichen Kriterien wie für Variablennamen. (Also erstes Zeichen alphabetisch, alle weiteren alphanumerisch, nur die ersten beiden Zeichen werden vom Basicinterpreter berücksichtigt)

Weiterhin muß im Funktionsnamen gekennzeichnet werden, von welchem Typ das Ergebniss der Funktion sein soll. Die erfolgt wiederum nach den gleichen Kriterien, wie sie schon von den Variablennamen her bekannt sind. (Also % für ganzzahlige, ! für einfachgenaue, # für doppeltgenaue und \$ für String-Ergebnisse. Weglassen ergibt einfachgenaue Ergebnisse.)

VARLIST besteht aus sovielen Variablen (jeweils durch Komma getrennt), wie der Ausdruck auf der rechten Seite an Argumenten benötigt.

AUSDRUCK ist die Funktion selbst. Sie darf nur eine Anweisung von einer Zeile Länge sein. (Durch Doppelpunkt getrennte Anweisungen sind nicht erlaubt.)

Zum besseren Verständnis folgende Beispiele:

```
10 DEFFN CUBIC (X) = X*X*X
20 INPUT A
30 B = FN CUBIC (A)
40 PRINT B
```

In 10 wird die einfache Funktion CUBIC definiert. Sie liefert ein Ergebniss einfacher Genauigkeit und benötigt nur eine Variable in der Variablenliste, da sie nur ein Argument (X) im Ausdruck hat.

In 30 wird die Funktion CUBIC aufgerufen. Hier ist das Argument die Variable A. (Es könnte dort aber auch jeder beliebige Ausdruck einfacher Genauigkeit stehen.)

```
Oder: 10 DEFFN TE$ (X$,Y$) = X$+"/" +Y$
20 INPUT "VORWAHL";V$
30 INPUT "RUFNUMMER";R$
40 G$ = FN TE$ (V$,R$)
50 PRINT G$
```

Hier wird eine Funktion mit zwei Stringargumenten definiert, welche in diesem Fall auch wieder ein Stringergebniss liefert. (Was nicht immer der Fall sein muß!)

LINE INPUT (ZEILENEINGABE)

Mit dem normalen INPUT Befehl ist es nicht möglich in eine Stringvariablen Kommata oder Anführungsstriche einzugeben. Außerdem werden eingegebene, führende Leerzeichen nicht in die Stringvariable übernommen. Weiterhin wurde jede Eingabe mittels Ausgabe eines '?' aufgefordert. Dadurch verbot sich die Anwendung des INPUT Befehls in vielen Fällen. (Insbesondere Textverarbeitungsprogramme in Basic!)

LINE INPUT umgeht diese Probleme und hat folgendes Format:

```
LINE INPUT "TEXT"; VAR$
```

mit: TEXT ist ein beliebiger Text, welcher vor der Eingabe auf den Bildschirm geschrieben wird. (Kann einschließlich nachfolgendem Semikolon entfallen!)

VAR\$ ist die Stringvariable, welche die Eingabe übernehmen soll.

Kurzes Beispiel:

```
10 LINE INPUT "IHRE EINGABE BITTE!"; A$  
20 PRINT A$
```

Hier erscheint nach der Ausgabe des Textes nur der Cursor in der nächsten Zeile. (kein Fragezeichen mehr!)
Sie können nun beliebige Zeichen eingeben (incl. führender Leerzeichen, Kommata, Anführungsstriche usw.) bis Sie mittels <ENTER> die Eingabe abschließen.

MID\$ (ALS STATEMENT)

MID\$ als Statement ist eine Art Umkehrung der MID\$ Funktion. Mit dem MID\$ Statement wird ein Teil eines Strings durch einen anderen String ersetzt.

Das Format ist wie folgt:

```
MID$(VAR$,n1,n2)=AUSDRUCK$
```

mit: VAR\$ ist die zu verändernde Stringvariable.

n1 ist die Position, ab welcher die Veränderung in VAR\$ beignnen soll. (Erste Position im String ist 1!)

n2 gibt die Anzahl der zu verändernden Zeichen an. Kann auch entfallen. (einschl. vorstehendem Komma!) Dann wird AUDRUCK\$ soweit eingesetzt, daß VAR\$ ihre ursprüngliche Länge beibehält.

AUDRUCK\$ ist ein beliebiger Stringausdruck.

Beachten Sie, daß die Ersetzung immer nur soweit erfolgt, daß die Länge des zu verändernden String erhalten bleibt!

Für folgende Beispiele wird vorausgesetzt: A\$="ABCDEFGH"

| | | |
|------------------------|----------|----------------|
| MID\$(A\$,2,2)="12345" | liefert: | A\$="A12DEFGH" |
| MID\$(A\$,5) ="12345" | " | A\$="ABCD123" |
| MID\$(A\$,4,4)="" | " | A\$="ABCDEFGH" |

INSTR (STRINGSUCH FUNKTION)

Mit INSTR können Sie Teilstrings innerhalb eines String suchen.

INSTR hat folgendes Format:

INSTR(n,AUDRUCK1\$,AUDRUCK2\$)

mit: n ist die Position in AUDRUCK1\$, ab welcher gesucht werden soll. Kann entfallen. (einschl. nachfolgendem Komma) In diesem Fall wird ab der 1. Position gesucht.

AUDRUCK1\$ ist der Stringausdruck, in welchem gesucht werden soll.

AUDRUCK2\$ ist der Stringausdruck, nach welchem gesucht werden soll.

Der Aufruf der INSTR Funktion liefert als Ergebniss die Position, ab welcher der gesuchte String zuerst im zu durchsuchenden String auftritt. Wird der Teilstring nicht gefunden, so ergibt sich 0 als Ergebniss.

Bei den folgenden Beispielen wird wieder A\$="ABCDEFGH" vorausgesetzt:

| | | |
|-------------------|---------|---|
| INSTR(A\$,"EF") | liefert | 5 |
| INSTR(A\$,"234") | " | 0 |
| INSTR(3,A\$,"AB") | " | 0 |

RESTORE N

Mit RESTORE N ist direkter Zugriff auf jede beliebige DATA-Zeile möglich. N muß nicht wie bei GOTO, GOSUB, usw. eine Ganzzahlkonstante sein, sondern kann ein beliebiger Ausdruck sein. Die RESTORE-Anweisung ohne Zeilennummer ist natürlich auch weiterhin verfügbar.

Beispiel:

```
10 DATA 10
20 DATA 20
30 DATA 30
40 RESTORE: READ A: PRINT A;
50 RESTORE 30: READ A: PRINT A;
60 RESTORE 3*10: READ A: PRINT A;
```

ergibt beim Starten mit RUN:

```
10 30 30
```

NAME und LSET (formatiertes Listen)

Diese Befehle entsprechen in Funktion und Format dem LIST bzw. LLIST Befehl. Aber es erfolgt eine formatierte Ausgabe auf Bildschirm bzw. Drucker, das heißt:

- Text beginnt immer erst ab Spalte 6 (davor stehen nur Zeilennummern!)
- Vor und hinter jedem BASIC-Schlüsselwort wird ein Leerzeichen eingefügt
- Nach jedem ':' wird eine neue Ausgabezeile begonnen
- IF-THEN-ELSE Strukturen werden eingerückt

In dieser Form ausgegeben sind auch Programme, welche durch einen 'PACKER' gegangen sind, wieder akzeptabel lesbar. Formatierte Listings vom Drucker eignen sich ausgezeichnet als endgültige Dokumentationen von BASIC-Programmen.

KILL (PACKER)

Ein PACKER entfernt aus Basicprogrammen überflüssige Leerzeichen und Kommentare ohne an der Funktion des Programmes etwas zu ändern.

Der Aufruf erfolgt mit:

KILL < ZEILE < - ZEILE >>

Durch diesen Befehl werden im ganzen Programmtext (- oder bei Angabe der optionalen Parameter, im spezifizierten Bereich -) alle Leerzeichen und Kommentare entfernt.

Beim Packen des Programmes werden alle Vorsichtsmaßnahmen getroffen, um die Funktion des Programmes nicht zu beeinträchtigen:

1) Reine Kommentarzeilen werden nur dann vollständig entfernt, wenn sie nicht ein Sprungziel bilden. Andernfalls bleiben Zeilennummer und REM erhalten.

```
10 GOTO 30
20 REM *****
30 REM ENDE
```

```
10 GOTO30
30 REM
```

2) In Hochkommata eingeschlossene Zeichenketten werden vom Packen nicht tangiert.

```
10 PRINT "EINGABE  ";
```

```
10 PRINT"EINGABE  ";
```

3) Nicht in Hochkommata eingeschlossene DATA-Strings werden so gepackt, daß sich der durch READ gelesene String nicht ändert.

```
10 DATA 1 , 2 , 3 , 4
```

```
10 DATA1 ,2 ,3 ,4
```

In seltenen Fällen, die von PACK nicht erkannt werden, kann das Packen unter ungünstigen Voraussetzungen zu einer Änderung der Funktion des Programmes führen. Ein Beispiel soll das Problem verdeutlichen:

```
10 IF A=F OR I=10 THEN 100
wird zu
10 IFA=FORI=10THEN100
```

gepackt. Wird die Zeile 10 nach dem Packen editiert und dadurch neu in den Zwischencode übersetzt, dann wird das Keyword FOR erkannt und eine Ausführung der Zeile ergäbe einen Syntax-Error.

Vor und nach dem Packen wird der Speicherbedarf des Programmes ausgegeben, um eine Abschätzung des eingesparten Speicherplatzes zu ermöglichen.

```

00100 ;*****
00110 ;***** COLBAS VERSION 1.0 *****
00120 ;***** 21.10.83 *****
00130 ;*****
00140 ;
00150 ;
00160 ;
00170 ;
00180 ;
00190 ;ASEG
00200 ;
00210 ;***** INITIALISIERUNG DER BASIC ZEIGER DURCH DIREKTES ÜBERSCHREIBEN
00220 ;
00230 ;
4155 00240 ORG 4155H
4155 C35FB7 00250 JP FN
00260 ;
415B 00270 ORG 415BH
415B C3A8B6 00280 JP DEF
00290 ;
417F 00300 ORG 417FH
417F C35ABC 00310 JP GET
00320 ;
418B 00330 ORG 418BH
418B C350B8 00340 JP MERGE
00350 ;
419A 00360 ORG 419AH
419A C37DB8 00370 JP RSET
419D C333B9 00380 JP INSTR
00390 ;
41A3 00400 ORG 41A3H
41A3 C309B9 00410 JP LINE
00420 ;
418E 00430 ORG 418EH
418E C3D8B9 00440 JP FLIST ;MITTELS "NAME" LIST AUF BILDSCHIRM
00450 ;
4197 00460 ORG 4197H
4197 C3E7B9 00470 JP FLLIST ;MITTELS "LSET" LIST AUF DRUCKER
00480 ;
4191 00490 ORG KILLV
4191 C3DABD 00500 JP KILL
00510 ;
00520 ;
00530 ;
00540 ;
00550 ;
00560 ;***** INITIALISIEREN DER BASICZEIGER PER PROGRAMM
00570 ;
00580 ;
B67C 00590 ORG 0B67CH
00600 ;
B67C 2A0440 00610 INIT LD HL, (RST10V)
B67F 22B6B9 00620 LD (CHV10+1), HL
B682 21A5B9 00630 LD HL, RESTN
B685 220440 00640 LD (RST10V), HL ;RST 10H-VEKTOR AENDERN
00650 ;
B688 3EC3 00660 LD A, 0C3H ;OPCODE FUER JP NACH A
00670 ;
B68A 32A941 00680 LD (41A9H), A
B68D 210DB7 00690 LD HL, USR

```

```

B690 22AA41 00700 LD (41AAH),HL
00710 ;
B693 22AC41 00720 LD (41ACH),A
00730 ;
00740 ;
00750 ;
00760 ;
B696 2126B8 00770 LD HL,READY
B699 22AD41 00780 LD (41ADH),HL
00790 ;
B69C 32D941 00800 LD (41D9H),A
B69F 2193B9 00810 LD HL,MID
B6A2 22DA41 00820 LD (41DAH),HL
00830 ;
B6A5 C3191A 00840 JF 1A19H
00850 ;
00860 ;
00870 ;
00880 ;
00890 ;
00900 ;
00910 ;CSEG
00920 ;
00930 ;***** PROGRAMMIERFÜR
00940 ;
00950 ;
00960 ;
013B 00970 SNERR EQU 013BH ;ANSFRUNGADR. FÜR SN ERR
00980 ;
00990 ;
01000 ;***** DEFFN UND DEFUSR BEFEHLE
01010 ;
B6A8 FEC1 01020 DEF CF 0C1H
B6AA 282C 01030 JR Z,DEFUSR
01040 ;
B6AC CDCCB6 01050 DEFFN CALL DEFFN1
B6AF CD282B 01060 CALL 282BH
B6B2 EB 01070 EX DE,HL
B6B3 73 01080 LD (HL),E
B6B4 23 01090 INC HL
B6B5 72 01100 LD (HL),D
B6B6 EB 01110 EX DE,HL
B6B7 7E 01120 LD A,(HL)
B6B8 FE28 01130 CP 28H
B6BA C2051F 01140 JP NZ,1F05H
B6BD D7 01150 RST 16
B6BE CD0D26 01160 DEFFN2 CALL 260DH
B6C1 7E 01170 LD A,(HL)
B6C2 FE29 01180 CP 29H
B6C4 CA051F 01190 JF Z,1F05H
B6C7 CF 01200 RST 8
B6C8 2C 01210 DEFB ' '
B6C9 C3BEB6 01220 JP DEFFN2
B6CC CF 01230 DEFFN1 RST 8
B6CD BE 01240 DEFB 0BEH ;FN TOKEN
B6CE 3E80 01250 LD A,80H
B6D0 32DC40 01260 LD (40DCH),A
B6D3 B6 01270 OR (HL)
B6D4 47 01280 LD B,A
B6D5 C31226 01290 JP 2612H

```

```

01300 ;
B6D8 CDE7B6 01310 DEFUSR CALL DEFUS1
B6DB D5 01320 PUSH DE
B6DC CF 01330 RST 8
B6DD D5 01340 DEFB 0D5H ; '=' TOKEN
B6DE CD022B 01350 CALL 2B02H
B6E1 E3 01360 EX (SP),HL
B6E2 73 01370 LD (HL),E
B6E3 23 01380 INC HL
B6E4 72 01390 LD (HL),D
B6E5 E1 01400 POP HL
B6E6 C9 01410 RET
B6E7 D7 01420 DEFUS1 RST 16
B6E8 010000 01430 LD BC,0
B6EB 3005 01440 JR NC,DEFUS2
B6ED D630 01450 SUB 30H
B6EF 17 01460 RLA
B6F0 4F 01470 LD C,A
B6F1 D7 01480 RST 16
B6F2 EB 01490 DEFUS2 EX DE,HL
B6F3 21F9B6 01500 LD HL,STORE
B6F6 09 01510 ADD HL,BC
B6F7 EB 01520 EX DE,HL
B6F8 C9 01530 RET
01540 ;
B6F9 4A1E 01550 STORE DEFW 1E4AH ;FC ERROR ALS DEFAULTADR.
B6FB A41E 01560 DEFW 1EA4H
B6FD A41E 01570 DEFW 1EA4H
B6FF A41E 01580 DEFW 1EA4H
B701 A41E 01590 DEFW 1EA4H
B703 A41E 01600 DEFW 1EA4H
B705 A41E 01610 DEFW 1EA4H
B707 A41E 01620 DEFW 1EA4H
B709 A41E 01630 DEFW 1EA4H
B70B A41E 01640 DEFW 1EA4H
01650 ;
01660 ;
01670 ;***** USR FUNKTION
01680 ;
B70D F1 01690 USR POP AF
B70E CDE7B6 01700 CALL DEFUS1
B711 D5 01710 PUSH DE
B712 CD2C25 01720 CALL 252CH
B715 E3 01730 EX (SP),HL
B716 4E 01740 LD C,(HL)
B717 23 01750 INC HL
B718 46 01760 LD B,(HL)
B719 21E726 01770 LD HL,26E7H
B71C E5 01780 PUSH HL
B71D C5 01790 PUSH BC
B71E 3AAF40 01800 LD A,(40AFH)
B721 F5 01810 PUSH AF
B722 FE03 01820 CP 3
B724 CCDA29 01830 CALL 2,29DAH
B727 F1 01840 POP AF
B728 EB 01850 EX DE,HL
B729 212141 01860 LD HL,4121H
B72C C9 01870 RET
01880 ;
01890 ;

```

```

01900 ;***** FN FUNKTION
01910 ;
B72D 1E2E 01920 AS2 LD E, 2EH
B72F C3A219 01930 JP 19A2H
B732 3AAF40 01940 AS5 LD A, (40AFH)
B735 3D 01950 DEC A
B736 3D 01960 DEC A
B737 3D 01970 DEC A
B738 B7 01980 OR A
B739 37 01990 SCF
B73A CA93B7 02000 JP Z, AS6
B73D 4E 02010 LD C, (HL)
B73E 23 02020 INC HL
B73F 46 02030 LD B, (HL)
B740 C5 02040 PUSH BC
B741 FA5CB7 02050 JP M, AS7
B744 23 02060 INC HL
B745 4E 02070 LD C, (HL)
B746 23 02080 INC HL
B747 46 02090 LD B, (HL)
B748 C5 02100 PUSH BC
B749 E25CB7 02110 JP PO, AS7
B74C 23 02120 INC HL
B74D DA53B7 02130 JP C, AS8
B750 211D41 02140 LD HL, 411DH
B753 4E 02150 AS8 LD C, (HL)
B754 23 02160 INC HL
B755 46 02170 LD B, (HL)
B756 23 02180 INC HL
B757 C5 02190 PUSH BC
B758 4E 02200 LD C, (HL)
B759 23 02210 INC HL
B75A 46 02220 LD B, (HL)
B75B C5 02230 PUSH BC
B75C DA9DB7 02240 AS7 JP C, AS9
B75F CDCCB6 02250 FN CALL DEFFN1
B762 3AAF40 02260 LD A, (40AFH)
B765 B7 02270 OR A
B766 F5 02280 PUSH AF
B767 22F340 02290 LD (40F3H), HL
B76A EB 02300 EX DE, HL
B76B 7E 02310 LD A, (HL)
B76C 23 02320 INC HL
B76D 66 02330 LD H, (HL)
B76E 6F 02340 LD L, A
B76F B4 02350 OR H
B770 CA2DB7 02360 JP Z, AS2
B773 7E 02370 LD A, (HL)
B774 FE28 02380 CP 28H
B776 C2C6B7 02390 JP NZ, AS3+1
B779 D7 02400 RST 16
B77A 22D840 02410 LD (40D8H), HL
B77D 1802 02420 JR AS4
B77F CF 02430 AS10 RST 8
B780 2C 02440 INC L
B781 0E04 02450 AS4 LD C, 04
B783 CD6319 02460 CALL 1963H
B786 3E80 02470 LD A, 80H
B788 32DC40 02480 LD (40DCH), A
B78B CD0D26 02490 CALL 260DH

```

| | | | | |
|------|--------|------------|------|------------|
| B78E | EB | 02500 | EX | DE,HL |
| B78F | 37 | 02510 | SCF | |
| B790 | C332B7 | 02520 AS11 | JP | AS5 |
| B793 | D22324 | 02530 AS6 | JP | NC,2423H |
| B796 | D5 | 02540 | PUSH | DE |
| B797 | EB | 02550 | EX | DE,HL |
| B798 | CD8828 | 02560 | CALL | 2888H |
| B79B | D1 | 02570 | POP | DE |
| B79C | AF | 02580 | XOR | A |
| B79D | E5 | 02590 AS9 | PUSH | HL |
| B79E | F5 | 02600 | PUSH | AF |
| B79F | EB | 02610 | EX | DE,HL |
| B7A0 | 7E | 02620 | LD | A,(HL) |
| B7A1 | FE29 | 02630 | CP | 29H |
| B7A3 | 20DA | 02640 | JR | NZ,AS10 |
| B7A5 | 2AF340 | 02650 | LD | HL,(40F3H) |
| B7A8 | CF | 02660 | RST | 8 |
| B7A9 | 28E5 | 02670 | JR | Z,AS11 |
| B7AB | 2AD840 | 02680 | LD | HL,(40D8H) |
| B7AE | CD0D26 | 02690 AS13 | CALL | 260DH |
| B7B1 | E3 | 02700 | EX | (SP),HL |
| B7B2 | CD2B1F | 02710 | CALL | 1F2BH |
| B7B5 | 7E | 02720 | LD | A,(HL) |
| B7B6 | FE29 | 02730 | CP | 29H |
| B7B8 | 2807 | 02740 | JR | Z,AS12 |
| B7BA | CF | 02750 | RST | 8 |
| B7BB | 2C | 02760 | INC | L |
| B7BC | E3 | 02770 | EX | (SP),HL |
| B7BD | CF | 02780 | RST | 8 |
| B7BE | 2C | 02790 | INC | L |
| B7BF | 18ED | 02800 | JP | AS17 |
| B7C1 | D7 | 02810 AS12 | RST | 16 |
| B7C2 | E3 | 02820 | EX | (SP),HL |
| B7C7 | CF | 02830 | RST | 8 |
| B7C4 | 29 | 02840 | ADD | HL,HL |
| B7C5 | 3ED5 | 02850 AS3 | LD | A,0D5H |
| B7C7 | CF | 02860 | RST | 8 |
| B7C8 | D5 | 02870 | PUSH | DE |
| B7C9 | CD3723 | 02880 | CALL | 2337H |
| B7CC | 2B | 02890 | DEC | HL |
| B7CD | D7 | 02900 | RST | 16 |
| B7CE | C29719 | 02910 | JP | NZ,1997H |
| B7D1 | E7 | 02920 | RST | 32 |
| B7D2 | 282F | 02930 | JR | Z,AT1 |
| B7D4 | D1 | 02940 AT6 | POP | DE |
| B7D5 | F1 | 02950 AT4 | POP | AF |
| B7D6 | 283F | 02960 | JR | Z,AT2 |
| B7D8 | 301C | 02970 | JR | NC,AT3 |
| B7DA | E1 | 02980 | POP | HL |
| B7DB | C1 | 02990 | POP | BC |
| B7DC | 70 | 03000 | LD | (HL),B |
| B7DD | 2B | 03010 | DEC | HL |
| B7DE | 71 | 03020 | LD | (HL),C |
| B7DF | FAD5B7 | 03030 | JP | M,AT4 |
| B7E2 | 2B | 03040 | DEC | HL |
| B7E3 | C1 | 03050 | POP | BC |
| B7E4 | 70 | 03060 | LD | (HL),B |
| B7E5 | 2B | 03070 | DEC | HL |
| B7E6 | 71 | 03080 | LD | (HL),C |
| B7E7 | E2D5B7 | 03090 | JP | PO,AT4 |

```

B7EA 2B      03100      DEC      HL
B7EB C1      03110      POP      BC
B7EC 70      03120      LD       (HL),B
B7ED 2F      03130      DEC      HL
B7EE 71      03140      LD       (HL),C
B7EF 2B      03150      DEC      HL
B7F0 C1      03160      POP      BC
B7F1 70      03170      LD       (HL),B
B7F2 2B      03180      DEC      HL
B7F3 71      03190      LD       (HL),C
B7F4 18DF    03200      JR        AT4
B7F6 D5      03210 AT3    PUSH    DE
B7F7 F5      03220      PUSH    AF
B7F8 E7      03230      RST     32
B7F9 11D340  03240      LD       DE,40D3H
B7FC CC8828  03250      CALL    Z,2888H
B7FF F1      03260      POP      AF
B800 C31A28  03270      JP       281AH
B803 2AB340  03280 AT1    LD       HL,(40B3H)
B806 EF      03290      EX       DE,HL
B807 2A2141  03300      LD       HL,(4121H)
B80A DF      03310      RST     24
B80B 3805    03320      JR        C,AT5
B80D CD4328  03330      CALL    2843H
B810 18C2    03340      JR        AT6
B812 D1      03350 AT5    POP      DE
B813 21D340  03360      LD       HL,40D3H
B816 E5      03370      PUSH    HL
B817 CDF529  03380 AT2    CALL    29F5H
B81A 7E      03390      LD       A,(HL)
B81B 22B340  03400      LD       (40B3H),HL
B81E E1      03410      POP      HL
B81F 77      03420      LD       (HL),A
B820 23      03430      INC      HL
B821 71      03440      LD       (HL),C
B822 23      03450      INC      HL
B823 70      03460      LD       (HL),B
B824 18AF    03470      JR        AT4
          03480 ;
          03490 ;
          03500 ;**** NEUE READY-MELDUNG
          03510 ;
B826 CDF920  03520 READY  CALL    20F9H
B829 2133B8  03530      LD       HL,611
B82C CD9238  03540      CALL    3892H
B82F E1      03550      POP      HL
B830 C32B1A  03560      JP       1A2BH
B833 43      03570 G11    DEFB    'COLBAS READY'
B83F 0D      03580      DEFB    0DH
B840 00      03590 G12    DEFB    0
B841 20      03600      DEFB    ' MERGE MODE #'
B84E 0D      03610      DEFB    0DH
B84F 00      03620      DEFB    0
          03630 ;
          03640 ;
          03650 ;**** MERGE BEFEHL
          03660 ;
B850 3A40B8  03670 MERGE  LD       A,(G12)
B853 B7      03680      OR       A
B854 2006    03690      JR        NZ,ME1

```



```

B856 2AA440 03700 LD HL, (40A4H)
B859 2291B8 03710 LD (BASBEG), HL
B85C ED5BF940 03720 ME1 LD DE, (40F9H)
B860 1B 03730 DEC DE
B861 D5 03740 PUSH DE
B862 C1 03750 POP BC
B863 1B 03760 DEC DE
B864 1B 03770 DEC DE
B865 1B 03780 DEC DE
B866 1A 03790 LD A, (DE)
B867 B7 03800 OR A
B868 2005 03810 JR NZ, ME2
B86A ED43F940 03820 LD (40F9H), BC
B86E 1B 03830 DEC DE
B86F 13 03840 ME2 INC DE
B870 13 03850 INC DE
B871 ED53A440 03860 LD (40A4H), DE
B875 3E2A 03870 LD A, 42
B877 3240B8 03880 LD (612), A
B87A C316BD 03890 JP RETURN

```

```
03900 ;
```

```
03910 ;
```

```
03920 ;***** RSET BEFEHL
```

```
03930 ;
```

```

B87D 3A40B8 03940 RSET LD A, (612)
B880 B7 03950 OR A
B881 180F 03960 JR Z, RS1
B883 2A91B8 03970 LD HL, (BASBEG)
B886 22A440 03980 LD (40A4H), HL
B889 7E00 03990 LD A, 0
B88B 3240B8 04000 LD (612), A
B88E C316BD 04010 RS1 JP RETURN

```

```
04020 ;
```

```
0002 04030 BASBEG DEFS 2
```

```
04040 ;
```

```
04050 ;
```

```
04060 ;
```

```
04070 ;
```

```
04080 ;
```

```
04090 ;***** MIDE ALS STATEMENT
```

```
04100 ;
```

```

B893 D7 04110 MIDE RST 10H
B894 CF 04120 RST 08H
B895 2B 04130 DEFM '...'
B896 ED03B8 04140 CALL MIDE
B899 E5 04150 PUSH HL
B89A D5 04160 PUSH DE
B89B EB 04170 EX DE, HL
B89C 23 04180 INC HL
B89D 5E 04190 LD E, (HL)
B89E 23 04200 INC HL
B89F 56 04210 LD D, (HL)
B8A0 2AA040 04220 LD HL, (40A0H)
B8A3 DF 04230 RST 18H
B8A4 E1 04240 POP HL
B8A5 E5 04250 PUSH HL
B8A6 D4FBB8 04260 CALL NC, MIDE
B8A9 E1 04270 POP HL
B8AA E3 04280 EX (SP), HL
B8AB CF 04290 RST 08H

```

```

B8AC 2C      04300      INC      L
B8AD CD1C2B  04310      CALL     2B1CH
B8B0 B7      04320      OR       A
B8B1 CA4A1E  04330      JP       Z,1E4AH
B8B4 F5      04340      PUSH     AF
B8B5 7E      04350      LD       A,(HL)
B8B6 FE2C    04360      CP       2CH
B8B8 3EFF    04370      LD       A,OFFH
B8BA 2004    04380      JR       NZ,MID1
B8BC D7      04390      RST      10H
B8BD CD1C2B  04400      CALL     2B1CH
B8C0 F5      04410 MID1  PUSH     AF
B8C1 CF      04420      RST      08H
B8C2 29      04430      DEFM    ' ) '
B8C3 CF      04440      RST      08H
B8C4 D5      04450      DEFB    0D5H
B8C5 CD3723  04460      CALL     2337H
B8C8 E5      04470      PUSH     HL
B8C9 CDD729  04480      CALL     29D7H
B8CC D1      04490      POP      DE
B8CD C1      04500      POP      BC
B8CE F1      04510      POP      AF
B8CF 4F      04520      LD       C,A
B8D0 EB      04530      EX       DE,HL
B8D1 E3      04540      EX       (SP),HL
B8D2 D5      04550      PUSH     DE
B8D3 7E      04560      LD       A,(HL)
B8D4 23      04570      INC      HL
B8D5 5E      04580      LD       E,(HL)
B8D6 23      04590      INC      HL
B8D7 56      04600      LD       D,(HL)
B8D8 69      04610      LD       L,C
B8D9 2600    04620      LD       H,0
B8DB 19      04630      ADD      HL,DE
B8DC 2B      04640      DEC      HL
B8DD 91      04650      SUB      C
B8DE DA4A1E  04660      JP       C,1E4AH
B8E1 3C      04670      INC      A
B8E2 B8      04680      CP       B
B8E3 3001    04690      JR       NC,MID2
B8E5 47      04700      LD       B,A
B8E6 E3      04710 MID2  EX       (SP),HL
B8E7 7E      04720      LD       A,(HL)
B8E8 23      04730      INC      HL
B8E9 5E      04740      LD       E,(HL)
B8EA 23      04750      INC      HL
B8EB 56      04760      LD       D,(HL)
B8EC E1      04770      POP      HL
B8ED B8      04780      CP       B
B8EE 3801    04790      JR       C,MID3
B8F0 78      04800      LD       A,B
B8F1 CDF6B8  04810 MID3  CALL     MID4
B8F4 E1      04820      POP      HL
B8F5 C9      04830      RET
          04840 ;
          04850 ;UNTERPROGRAMME
          04860 ;
B8F6 B7      04870 MID4  OR       A
B8F7 C2D609  04880      JP       NZ,09D6H
B8FA C9      04890      RET

```

; "=" TOKEN

```

B8FB E5      04900 MID5      PUSH      HL
B8FC CD4328   04910          CALL      2843H
B8FF E1      04920          POP       HL
B900 CD3309   04930          JP       09D3H
B903 CD0D26   04940 MID6      CALL      280DH
B906 C3F40A   04950          JP       0AF4H
              04960 ;
              04970 ;
              04980 ;
              04990 ;**** LINE INPUT BEFEHL
              05000 ;
B909 CF      05010 LINE      RST       08H
B90A 89      05020          DEFB      89H          ; "INPUT"-TO EN
B90B CD2828   05030          CALL      2828H
B90E 7E      05040          LD        A, (HL)
B90F FE23     05050          CP        '#'
B911 CA3B01   05060          JP       Z, SNEFR      ; ERFOR AUSSPRUNG
B914 CDCD21   05070          CALL      21CDH
B917 CD0D26   05080          CALL      280DH
B91A CDF40A   05090          CALL      0AF4H
B91D E5      05100          PUSH      HL
B91E D5      05110          PUSH      DE
B91F CD6103   05120          CALL      0361H
B922 380A     05130          JP       C, LINE1
B924 0600     05140          LD        B, 0
B926 CD6828   05150          CALL      2868H
B929 E1      05160          POP       HL
B92A AF      05170          XOR       A
B92B C3331F   05180          JP       1F33H
B92E F1      05190 LINE1     POP       AF
B92F F1      05200          POP       AF
B930 C3BE1D   05210          JP       1DBEH
              05220 ;
              05230 ;
              05240 ;
              05250 ;**** INSTR BEFEHL
              05260 ;
B933 D7      05270 INSTR     RST       10H
B934 CD3523   05280          CALL      2335H
B937 E7      05290          RST       20H
B938 3E01     05300          LD        A, 01H
B93A C41F2B   05310          CALL      NZ, 2B1FH
B93D B7      05320          OR        A
B93E CA4A1E   05330          JP       Z, 1E4AH
B941 F5      05340          PUSH      AF
B942 E7      05350          RST       20H
B943 C4A0B9   05360          CALL      NZ, INSTR8
B946 CDF40A   05370          CALL      0AF4H
B949 ED4B2141 05380          LD        BC, (4121H)
B94D C5      05390          PUSH      BC
B94E CDA0B9   05400          CALL      INSTR8
B951 CF      05410          RST       08H
B952 29      05420          DEFM      ')'
B953 E3      05430          EX        (SP), HL
B954 E5      05440          PUSH      HL
B955 CD0729   05450          CALL      29D7H
B958 E7      05460          EX        (SP), HL
B959 CDDD29   05470          CALL      29DDH
B95C D1      05480          POP       DE
B95D C1      05490          POP       BC

```

```

B95E F1      05500      POP      AF
B95F C5      05510      PUSH     BC
B960 F5      05520      PUSH     AF
B961 D5      05530      PUSH     DE
B962 3D      05540      DEC       A
B963 46      05550      LD        B, (HL)
B964 23      05560      INC       HL
B965 5E      05570      LD        E, (HL)
B966 23      05580      INC       HL
B967 56      05590      LD        D, (HL)
B968 6F      05600      LD        L, A
B969 2600    05610      LD        H, 0
B96B 19      05620      ADD      HL, DE
B96C E3      05630      EX       (SP), HL
B96D 4E      05640      LD        C, (HL)
B96E 23      05650      INC       HL
B96F 5E      05660      LD        E, (HL)
B970 23      05670      INC       HL
B971 56      05680      LD        D, (HL)
B972 E1      05690      POP      HL
B973 81      05700      ADD      A, C
B974 381D    05710      JR        C, INSTR4
B976 ED44    05720 INSTR1  NEG
B978 80      05730      ADD      A, B
B979 47      05740      LD        B, A
B97A 3017    05750      JR        NC, INSTR4
B97C 04      05760      INC       B
B97D E5      05770 INSTR2  PUSH     HL
B97E D5      05780      PUSH     DE
B97F C5      05790      PUSH     BC
B980 0C      05800      INC       C
B981 0D      05810 INSTR3  DEC       C
B982 2813    05820      JR        Z, INSTR5
B984 1A      05830      LD        A, (DE)
B985 BE      05840      CP       (HL)
B986 13      05850      INC       DE
B987 23      05860      INC       HL
B988 28F7    05870      JR        Z, INSTR3
B98A C1      05880      POP      BC
B98B D1      05890      POP      DE
B98C E1      05900      POP      HL
B98D F1      05910      POP      AF
B98E 3C      05920      INC       A
B98F 23      05930      INC       HL
B990 F5      05940      PUSH     AF
B991 10EA    05950      DJNZ     INSTR2
B993 F1      05960 INSTR4  POP      AF
B994 AF      05970      XOR      A
B995 1804    05980      JR        INSTR6
B997 C1      05990 INSTR5  POP      BC
B998 D1      06000      POP      DE
B999 E1      06010      POP      HL
B99A F1      06020      POP      AF
B99B CDF827  06030 INSTR6  CALL    27F8H
B99E E1      06040 INSTR7  POP      HL
B99F C9      06050      RET
B9A0 CF      06060 INSTR8  RST      08H
B9A1 2C      06070      DEFM    ', '
B9A2 C33723  06080      JP       2337H
          06090 ;

```

```

06100 ;
06110 ;
06120 ;
06130 ; ** RESTORE N **, 18.2.80 (V 1.1)
06140 ; BEARBEITET FUER EDTASM, 4.9.81
06150 ;
06160 ; BASIC-UNTERPROGRAMME:
1D78 06170 CHRGE1 EQU 1D78H ; ZEICHENHOLROUTINE
2B01 06180 INTEXP EQU 2B01H ; ZEILENNUMMER ERMITTELN
1B2C 06190 SLINE EQU 1B2CH ; ZEILE IN RAM SUCHEN
06CC 06200 WSTART EQU 06CCH ; ZURUECK ZUM BASIC
1D1E 06210 EXEC EQU 1D1EH ; BASICPROGRAMM AUSFUEHREN
06220 ;
06230 ; BASIC-ZEIGER:
40FF 06240 DATAZ EQU 40FFH ; DATAZEIGER
4004 06250 RST10V EQU 4004H ; RST 10H-VEKTOR
06260 ;
06270 ;
06280 ;
06290 ; RESTORE N
06300 ;
B9A5 E3 06310 RESTN EX (SP),HL ; AUFRUF VON 1DXXH?
B9A6 7C 06320 LD A,H
B9A7 FE1D 06330 CP 1DH
B9A9 2810 06340 JR Z,RESTN1 ; NEIN, RST 10H AUSF.
B9AB FE20 06350 CP 20H
B9AD 2005 06360 JR NZ,NINT
B9AF 7D 06370 LD A,L
B9B0 FE50 06380 CP 50H
B9B2 2807 06390 JR Z,RESTN1
B9B4 E3 06400 NINT EX (SP),HL
B9B5 C30000 06410 CHV10 JP 0
B9B8 E1 06420 NINT1 POP HL
B9B9 18FA 06430 JR CHV10
B9BB E3 06440 RESTN1 EX (SP),HL
B9BC E5 06450 PUSH HL
B9BD D7 06460 RST 10H
B9BE FE90 06470 CP 90H ; =RESTORE?
B9C0 20F6 06480 JR NZ,NINT1 ; NEIN, RST 10H OHNE INC HL
B9C2 C1 06490 POP BC
B9C3 D7 06500 RST 10H
B9C4 CA911D 06510 JP Z,1D91H
B9C7 CD022B 06520 CALL INTEXP+1 ; ZNP DECODIEREN
B9CA E5 06530 PUSH HL ; POINTER SETZEN
B9CE CD2C1B 06540 CALL SLINE ; ZEILENADRESSE
B9CE 0B 06550 DEC BC
B9CF ED43FF40 06560 LD (DATAZ),BC ; NACH DATAZEIGER
B9D3 E1 06570 POP HL ; POINTER ZURUECK
B9D4 C1 06580 POP BC ; RUECK SPRUNGADR.
B9D5 C31E1D 06590 JP EXEC ; NAECHSTE ; ZEILENADRESSE
06600 ; ***** FORMATIERTES LISTING FUR COLOUR-GENIE *****
06610 ; ***** VERSION 1.0 04.11.83 *****
06620 ; ***** (C) 1983 BY M. BUNING *****
06630 ;
06640 ; ***** ROM-ADRESSEN
06650 ;
1B10 06660 LARG EQU 1B10H ; LIST ARGUMENTE AUSWERTEN
20FE 06670 CR EQU 20FEH ; GIBT CR AUS
032A 06680 OUTCHF EQU 032AH ; GIBT ZEICHEN IN A AUS
1A18 06690 HAUPTS EQU 1A18H ; BASIC HAUPTSCHLEIFE

```

```

0FAF      06700 PRZNR   EQU      0FAFH      ;ZEILENNUMMER IN HL AUSGEBEN
1D9B      06710 BREAL   EQU      1D9BH      ;TASTENDRUCKE AUSWERTEN
39AD      06720 TABANF   EQU      39ADH      ;KEYWORDTABELLENANFANG NACH HL
          06730 ;
          06740 ;
          06750 ;
          06760 ;
          06770 ;**** SYSTEM-RAM-ADRESSEN
          06780 ;
40A7      06790 BUFANF   EQU      40A7H      ;ENTHAELT DEN ZEIGER AUF DEN BUFFERANF.
409C      06800 PVFLAG   EQU      409CH      ;FLAG (0=BILDSCHIRM, 1=DRUCKER)
40A6      06810 VPOS     EQU      40A6H      ;BILDSCHIRM-SPALTEN-POSITION
409B      06820 PPOS     EQU      409BH      ;DRUCKER-SPALTEN-POSITION
40A2      06830 AITZNR   EQU      40A2H      ;SPEICHERZELLEN FÜR AKTUELLE ZNR
40EC      06840 REFZNR   EQU      40ECH      ;SPEICHERZELLEN FÜR '.' BEZUG
          06850 ;
          06860 ;
          06870 ;
          06880 ;
          06890 ;**** BENUTZTE TOKEN
          06900 ;
008F      06910 IF       EQU      08FH; IF-TOKEN
00FB      06920 REMAP    EQU      0FBH; APOSTROPH REMARK
0000      06930 ZENDE    EQU      00H; ZEILENENDE
0081      06940 FCOLOUR   EQU      081H; FCOLOUR
00FF      06950 DOPTOK   EQU      0FFH; ERSTES BYTE EINES DOPPELTOKEN
          06960 ;
          06970 ;
          06980 ;**** START DER FLIST-ROUTINE
          06990 ;
B9D8 DD214FBC 07000 FLIST LD      IX,VARADR   ;IX ZEIGT AUF DEN VARIABLENBEREICH
B9DC DD360327 07010      LD      (IX+3),39   ;LETZTE SPALTE AUF BILDSCHIRM
B9E0 DD360603 07020      LD      (IX+6),3;MAXIMALE EINRÜCKUNG AUF BILDSCHIRM
B9E4 C3F8B9   07030      JF      FLIST1      ;EINSPIGUNG IN GEMEINSAMEN TEIL
          07040 ;
          07050 ;
          07060 ;**** START DER FLLIST-ROUTINE
          07070 ;
B9E7 DD214FBC 07080 FLLIST LD      IX,VARADR   ;IX ZEIGT AUF VARIABLENBEREICH
B9EB DD36034F 07090      LD      (IX+3),79   ;LETZTE SPALTE AUF DRUCKER
B9EF DD36060A 07100      LD      (IX+6),10   ;MAXIMALE EINRÜCKUNG AUF DRUCKER
B9F3 3E01     07110      LD      A,1 ;1=DRUCKER
B9F5 329C40   07120      LD      (PVFLAG),A  ;AUSGABE AUF DRUCKER SCHALTEN
          07130 ;
          07140 ;
          07150 ;
          07160 ;
          07170 ;**** GEMEINSAMER TEIL FÜR FORMATIERTES LIST U. LLIST
          07180 ;
B9FB C1       07190 FLIST1 POP     BC      ;RUCH SPRUNGADRESSE ENTFERNEN
B9F9 CD101B   07200      CALL    LARG;1. ZEILENADR. NACH BC, 2. ZNR AUF STACK
B9FC C5       07210      PUSH    BC      ;1. ZEILENADRESSE RETTEN
B9FD 21FFFF   07220 NEXTL LD      HL,0FFFFH  ;DIRECTMODE SETZEN
BA00 22A240   07230      LD      (AITZNR),HL ;UND IN ENTSPRECHENDEN ZELLEN ABSPEICHERN
BA03 E1       07240      POP     HL      ;1. ZEILENADRESSE NACH HL
BA04 D1       07250      POP     DE      ;2.ZNR NACH DE
BA05 4E       07260      LD      C,(HL)    ;ZEILENZEIGER LADEN
BA06 23       07270      INC     HL
BA07 46       07280      LD      B,(HL)    ;(MSB)
BA08 23       07290      INC     HL

```

```

BA09 78      07300      LD      A,B
BA0A B1      07310      OR       C ;PROGRAMMENDE ERREICHT?
BA0B CA191A  07320      JP       Z,HAUPTS+1 ;FALLS JA, DANN ZUR HAUPTSCHLEIFE
                07330 ;
BA0E CD9B1D  07340      CALL     BREAK ;BEHANDLUNG VON TASTENDRUECKEN
BA11 C5      07350      PUSH     BC ;NAECHSTE ZEILENADRESSE RETTEN
BA12 4E      07360      LD       C,(HL) ;AKTUELLE ZNR NACH BC (LSB)
BA13 23      07370      INC      HL
BA14 46      07380      LD       B,(HL) ;(MSB)
BA15 23      07390      INC      HL
BA16 C5      07400      PUSH     BC ;ZNR RETTEN
BA17 E3      07410      EX       (SP),HL ;ZEIGER RETTEN, ZNR NACH HL
BA18 EB      07420      EX       DE,HL ;2. ZNR NACH HL, ZNR NACH DE
BA19 DF      07430      RST      018H;ZNR 2. ZNR?
BA1A C1      07440      POP      BC ;ZEIGER NACH BC.
BA1B DA181A  07450      JP       C,HAUPTS;WENN 2. ZNR ERREICHT DANN HAUPTSCHLEIFE
BA1E E3      07460      EX       (SP),HL ;2. ZNR RETTEN, NAECHSTE ZEILENA. NACH HL
BA1F E5      07470      PUSH     HL ;NAECHSTE ZEILENADRESSE RETTEN
BA20 C5      07480      PUSH     BC ;ZEIGER RETTEN
BA21 EB      07490      EX       DE,HL ;AKTUELLE ZNR NACH HL
BA22 22EC40  07500      LD       (REFZNR),HL ;ALS '.' BEZUG ABSPEICHERN
BA25 CDAF0F  07510      CALL     PRZNR ;ZEILENNUMMER AUSGEBEN
                07520 ;
BA28 DD360000 07530      LD       (IX+0),0;INDENT-COUNTER NULL SETZEN
BA2C DD360500 07540      LD       (IX+5),0;THEN-ELSE-FLAG RUECK SETZEN
BA30 ED4BA740 07550      LD       BC,(BUFANF) ;BC ZEIGT AUF TEXTBUFFER-ANFANG
BA34 16FF     07560      LD       D,255 ;TEXTBUFFER-LAENGE AUF 255 SETZEN
BA36 E1      07570      POP      HL ;ZEIGER VOM STACK HOLEN
BA37 2B      07580      DEC      HL ;ZEIGER VOR 1. TEXTZEICHEN SETZEN
                07590 ;
BA38 D7      07600      RST      10H ;LEERZEICHEN AM ZEILENANFANG UBERGEHEN
BA39 FE3A     07610      CP       ':' ;IST 1. RELEVANTES ZEICHEN ':'?
BA3B 2001     07620      JR       NZ,FLIST2 ;FALLS NEIN, DANN ZEICHEN AUSWERTEN
BA3D D7      07630 FLIST2 RST      10H ;FALLS JA, DANN ':' UBERGEHEN
BA3E FE00     07640 FLIST3 CP       ZENDE ;IST ZEILENENDE-ZEICHEN ERREICHT
BA40 CA12BB   07650      JP       Z,LINEND;FALLS JA, DANN AUSSPRUNG
BA43 FE3A     07660      CP       ':' ;IST ZEICHEN ':'?
BA45 200A     07670      JR       NZ,FLIST5 ;FALLS NEIN, DANN WEITERE AUSWERTUNG
                07680 ;
BA47 D7      07690 FLIST4 RST      10H ;NAECHSTES RELEVANTES ZEICHEN SUCHEN
BA48 FE3A     07700      CP       ':' ;IST ES WIEDER ':'?
BA4A 28FB     07710      JR       Z,FLIST4;FALLS JA, DANN ':' UBERGEHEN
BA4C CD38BB   07720      CALL     PRINT ;FALLS NEIN, DANN TEXTBUFFER AUSGEBEN
BA4F 18ED     07730      JR       FLIST3 ;UND ZEILE WEITER ABARBEITEN
                07740 ;
BA51 A7      07750 FLIST5 AND      A ;FLAGS GEMAESS A SETZEN
BA52 F2FFBA   07760      JP       P,NOTOF ;FALLS KEIN TOFEN, SPRUNG NACH NOTOF
                07770 ;
                07780 ;
                07790 ;
                07800 ;
                07810 ;***** TOFEN AUSWERTEN
                07820 ;
BA55 FE93     07830 TOF CP       REM ;REM-TOFEN?
BA57 2015     07840      JR       NZ,TOF3 ;FALLS NICHT, WEITER AUSWERTEN
                07850 ;
BA59 CDF9BB   07860      CALL     TOFBUF ;'REM' AUSGEBEN
BA5C 23      07870      INC      HL ;HL ZEIGT AUF NAECHSTES ZEICHEN
BA5D 7E      07880      LD       A,(HL) ;ZEICHEN NACH A BRINGEN
BA5E FEFB     07890      CP       REMAF ;IST ZEICHEN REMARK-APOSTROPH?

```

```

BA60 2002      07900      JR      NZ,TOF 2 ;FALLS NEIN, DANN WEITERBEARBEITEN
BA62 23        07910 TOF 1 INC      HL ;HL ZEIGT AUF NAECHSTES ZEICHEN
BA63 7E        07920      LD      A,(HL) ;ZEICHEN NACH A BRINGEN
BA64 FE00      07930 TOF 2 CP      ZENDE ;ZEILENENDE ERREICHT?
BA66 CA12BB    07940      JP      Z,LINEND;FALLS JA, DANN AUSSPRUNG
BA69 CD2DBB    07950      CALL    INBUF ;FALLS NEIN, DANN ZEICHEN NACH BUFFER
BA6C 1BF4      07960      JR      TOK1;NAECHSTES ZEICHEN BEARBEITEN
                07970 ;
BA6E FE88      07980 TOF 3 CP      DATA;DATA-TOKEN?
BA70 201F      07990      JR      NZ,TOF 6 ;FALLS NEIN, DANN WEITERBEARBEITEN
                08000 ;
BA72 CDF9BB    08010      CALL    TOF BUF ;'DATA' AUSGEBEN
BA75 23        08020 TOF 4 INC      HL ;HL ZEIGT AUF NAECHSTES ZEICHEN
BA76 7E        08030      LD      A,(HL) ;ZEICHEN NACH A
BA77 FE22      08040      CP      ' "' ;ANFUHRUNGSSTRICHE?
BA79 2008      08050      JR      NZ,TOK 5 ;FALLS NEIN, ZEICHEN WEITER BEARBEITEN
BA7B CD18BB    08060      CALL    QUOTE ;TEXT HINTER DEN QUOTES BEARBEITEN
BA7E CA12BB    08070      JP      Z,LINEND;AUSSPRUNG FALLS ZERO-FLAG GESETZT
BA81 1BF2      08080      JR      TOF 4;SONST NAECHSTES ZEICHEN BEARBEITEN
BA83 FE00      08090 TOF 5 CP      ZENDE ;ZEILENENDE ERREICHT?
BA85 CA12BB    08100      JP      Z,LINEND;FALLS JA, AUSSPRUNG
BA88 FE3A      08110      CP      ':' ;':' ERREICHT?
BA8A 28BB      08120      JR      Z,FLIST4;MULTIPLE ':' AUSBLENDEN, BUFFER AUSGEBEN
BA8C CD2DBB    08130      CALL    INBUF ;ANSONSTEN ZEICHEN IN DEN TEXTBUFFER
BA8F 1BE4      08140      JR      TOF 4;NAECHSTES ZEICHEN BEARBEITEN
                08150 ;
BA91 FE8F      08160 TOF 6 CP      IF ;IF-TOKEN?
BA93 2011      08170      JR      NZ,TOF 7 ;FALLS NICHT, WEITER
BA95 DD3400    08180      INC      (IX+0) ;INDENT-COUNTER INC
BA98 DD360801  08190      LD      (IX+8),1;SET IF-INDENT-FLAG
BA9C DD360A00  08200      LD      (IX+10),0 ;RESET ELSE-FLAG
BAA0 DD360901  08210      LD      (IX+9),1;SET IF-FLAG
BAA4 1853      08220      JR      TOF 10 ;UND HINTER DIE TOKEN-AUSWERTUNG
                08230 ;
BAA6 FECA      08240 TOF 7 CP      THEN;THEN-TOKEN?
BAA8 2008      08250      JR      NZ,TOF 8 ;FALLS NICHT, WEITER
BAAA CD38BB    08260      CALL    PRINT ;BUFFER LEEFEN
BAAD DD360501  08270      LD      (IX+5),1;SET THEN-ELSE-FLAG
BAB1 DD360900  08280      LD      (IX+9),0;RESET IF-FLAG
                08290 ;
BAB5 DDCB0946  08300 TOF 8 BIT      0,(IX+9);IF-FLAG RUCKGESETZT?
BAB9 2825      08310      JR      Z,TOF 9 ;FALLS JA, WEITERAUSWERTUNG UBERGEHEN
BABB FE8E      08320      CP      OREN;BEFEHLS TOKEN?
BABD F2E0BA    08330      JP      P,TOF 9 ;FALLS NICHT, WEITERAUSWERTUNG UBERGEHEN
BAC0 CD38BB    08340      CALL    PRINT ;BUFFER LEEFEN
BAC3 3E54      08350      LD      A,'T'
BAC5 CD2DBB    08360      CALL    INBUF
BAC8 3E48      08370      LD      A,'H'
BACA CD2DBB    08380      CALL    INBUF
BACD 3E45      08390      LD      A,'E'
BACF CD2DBB    08400      CALL    INBUF
BAD2 3E4E      08410      LD      A,'N'
BAD4 CD2DBB    08420      CALL    INBUF
BAD7 7E        08430      LD      A,(HL) ;AKTUELLES TOKEN WIEDER NACH A
BAD8 DD360501  08440      LD      (IX+5),1;SET THEN-ELSE-FLAG
BADC DD360900  08450      LD      (IX+9),0;RESET IF-FLAG
                08460 ;
BAE0 FE95      08470 TOF 9 CP      ELSE;ELSE-TOKEN?
BAE2 2015      08480      JR      NZ,TOF 10;FALLS NICHT, WEITERES UBERGEHEN
BAE4 DD360501  08490      LD      (IX+5),1;SET THEN-ELSE-FLAG

```



```

BAE8 DD360900 08500      LD      (IX+9),0;RESET IF-FLAG
BAEC DDCB0A46 08510      BIT      0,(IX+10) ;ELSE-FLAG GESETZT?
BAF0 DD360A01 08520      LD      (IX+10),1 ;SET ELSE-FLAG
BAF4 2803      08530      JR      Z,TOF10 ;FALLS NICHT, WEITERE AUSFUEHRUNGEN UEBERD
BAF6 DD3500      08540      DEC      (IX+0) ;SONST INDENT-COUNTER DEC
                08550 ;
BAF9 CDF9BB      08560 TOF10    CALL    TOFBUF ;TOFEN DECODIEREN U. IN BUFFER SCHREIBEN
BAFC C33DBA      08570      JP      FLIST2 ;NAECHSTES ZEICHEN BEARBEITEN
                08580 ;
                08590 ;
                08600 ;
                08610 ;
                08620 ;***** NICHT-TOKEN AUSWERTEN
                08630 ;
BAFF FE22      08640 NOTOF    CP      '"' ;ANFUHRUNGSSTRICHE?
BB01 2008      08650      JR      NZ,NOTOF1 ;FALLS NEIN, WEITER AUSWERTEN
                08660 ;
BB03 CD18BB      08670      CALL    QUOTE ;ZEICHEN NACH QUOTE AUSWERTEN
BB06 280A      08680      JR      Z,LINEND;FALLS ZERO-FLAG GESETZT AUSSPRUNG
BB08 C33DBA      08690      JP      FLIST2 ;SONST NAECHSTES ZEICHEN AUSWERTEN
                08700 ;
BB0B 7E      08710 NOTOF1    LD      A,(HL) ;AKTUELLES ZEICHEN NACH A
BB0C CD2DBB      08720      CALL    INBUF ;ZEICHEN IN DEN TEXTBUFFER BRINGEN
BB0F C33DBA      08730      JP      FLIST2 ;UND NAECHSTES ZEICHEN AUSWERTEN
                08740 ;
                08750 ;
                08760 ;
                08770 ;
                08780 ;***** AUSSPRUNG BEI ZEILENENDE
                08790 ;
BB12 CD38BB      08800 LINEND    CALL    PRINT ;BUFFER AUSDRUCKEN
BB15 C3FDB9      08810      JP      NEXTL ;ZUM AUSWERTEN DER NAECHSTEN ZEILE
                08820 ;
                08830 ;
                08840 ;
                08850 ;
                08860 ;
                08870 ;
                08880 ;***** VOM HAUPTPROGRAMM ANGESPRUNGENE UNTERPROGRAMME
                08890 ;
                08900 ;
                08910 ;
                08920 ;***** QUOTE
                08930 ;***** (BEHANDELT TEXT IN ANFUHRUNGSSTRICHEN)
                08940 ;
BB18 CD2DBB      08950 QUOTE    CALL    INBUF ;ANFUHRUNGSSTRICHE IN DEN BUFFER BRINGEN
BB1B 23      08960      INC      HL ;AUF NAECHSTES ZEICHEN ZEIGEN
BB1C 7E      08970      LD      A,(HL) ;NACH A BRINGEN
BB1D FE00      08980      CP      ZENDE ;ZEILENENDE ERREICHT?
BB1F 2002      08990      JR      NZ,QUOTE1 ;FALLS NEIN, DANN WEITER ARBEITEN
                09000 ;
BB21 AF      09010      XOR      A ;FALLS JA, DANN END-FLAG (ZERO) SETZEN
BB22 C9      09020      RET      ;UND ZURUECK
                09030 ;
BB23 FE22      09040 QUOTE1    CP      '"' ;WIEDER ANFUHRUNGSSTRICHE?
BB25 20F1      09050      JR      NZ,QUOTE;FALLS NEIN, SCHLEIFE
BB27 CD2DBB      09060      CALL    INBUF ;FALLS JA, DANN QUOTES NACH BUFFER
BB2A AF      09070      XOR      A ;A=0
BB2B 3C      09080      INC      A ;A=1 (DAMIT ZERO-FLAG RUCH GESETZT)
BB2C C9      09090      RET      ;AUSSPRUNG MIT RUCH GESETZTEM END-FLAG

```

```

09100 ;
09110 ;
09120 ;
09130 ;
09140 ;***** INBUF
09150 ;***** (BRINGT ZEICHEN IN A IN DEN BUFFER, VERHINDERT
09160 ;***** UBERLAUFEN DES BUFFERS)
09170 ;
BB2D 02 09180 INBUF LD (BC),A ;SCHREIBT A GEMAESS BUFFERZEIGER EIN
BB2E DD7702 09190 LD (IX+2),A;NACH (IX+2) ALS LETZTES AUSGEGEBENES Z.
BB31 7A 09200 LD A,D ;VERBLEIBENDE BUFFERLAENGE NACH A
BB32 FE00 09210 CP 0 ;IST BUFFER VOLL?
BB34 C8 09220 RET Z ;FALLS JA, DANN RUCH SPRUNG
BB35 03 09230 INC BC ;FALLS NEIN, DANN BUFFERZEIGER INC.
BB36 15 09240 DEC D ;VERBLEIBENDE BUFFERLAENGE DEC.
BB37 C9 09250 RET ;UND RUCH SPRINGEN
09260 ;
09270 ;
09280 ;
09290 ;
09300 ;***** PRINT
09310 ;***** (GIBT TEXTBUFFER FORMATIERT AUS)
09320 ;
BB38 F5 09330 PRINT PUSH AF ;ZEICHEN IN A RETTEN
BB39 E5 09340 PUSH HL ;HL ALS ZEIGER RETTEN
BB3A AF 09350 XOR A ;A=0
BB3B CD2DBB 09360 CALL INBUF ;BUFFERINHALT MIT 0 ABSCHLIESSEN
BB3E 2AA740 09370 LD HL,(BUFANF) ;HL ZEIGT AUF BUFFERANFANG
BB41 7E 09380 PRINT1 LD A,(HL) ;ZEICHEN NACH A
BB42 FE00 09390 CP 0 ;BUFFER ENDE?
BB44 284A 09400 JR Z,PRINT8;FALLS JA, AUSSPRUNG
BB46 CDA5BB 09410 CALL INDCHI ;INDENT COUNTER AUFBEREITEN
BB49 3E20 09420 PRINT2 LD A,' ' ;LEERZEICHEN NACH A
BB4B CD2A03 09430 CALL OUTCHF ;UND AUSGEBEN
BB4E CDDDBB 09440 CALL PVPOS ;BILDSCHIRM ODER DRUCKERSPALTE HOLEN
BB51 DD7E01 09450 LD A,(IX+1);UND NACH A BRINGEN
BB54 FE06 09460 CP 6 ;6. SPALTE ERREICHT?
BB56 20F1 09470 JR NZ,PRINT2 ;FALLS NEIN, DANN SCHLEIFE
BB58 AF 09480 PRINT3 XOR A ;A=0
BB59 DDB604 09490 OR (IX+4) ;IST AUFBEREITETER INDENT COUNTER 0?
BB5C 2808 09500 JR Z,PRINT5;FALLS JA, WEITER
BB5E CDEFBB 09510 CALL BLANK5 ;SONST 5 LEERZEICHEN AUSGEBEN
BB61 DD3504 09520 DEC (IX+4) ;AUFBEREITETEN INDENT COUNTER DEC.
BB64 18F2 09530 JR PRINT3 ;UND SCHLEIFE
09540 ;
BB66 7E 09550 PRINT5 LD A,(HL) ;ZEICHEN NACH A
BB67 FE00 09560 CP 0 ;IST ENDE ERREICHT?
BB69 282E 09570 JR Z,PRINT9;FALLS JA, DANN ZUM AUSSPRUNG
BB6B CDDDBB 09580 CALL PVPOS ;SPALTENPOSITION NACH (IX+1) HOLEN
BB6E DD7E01 09590 LD A,(IX+1);UND NACH A BRINGEN
BB71 DDBE03 09600 CP (IX+3) ;IST A GLEICH MAX. SPALTENPOSITION?
BB74 2807 09610 JR Z,PRINT6;FALLS JA, NEUE ZEILE EINLEITEN
BB76 7E 09620 LD A,(HL) ;AKTUELLES ZEICHEN NACH A
BB77 CD2A03 09630 CALL OUTCHF ;UND ZEICHEN AUSGEBEN
BB7A 23 09640 INC HL ;ZEIGER AUF NAECHSTES ZEICHEN RUCHEN
BB7B 18E9 09650 JP PRINT5 ;UND SCHLEIFE
BB7D 3A9C40 09660 PRINT6 LD A,(PVFLAG) ;PRINTER-VIDEO FLAG LADEN
BB80 FE00 09670 CP 0 ;STEHT PV FLAG AUF BILDSCHIRM?
BB82 2805 09680 JR Z,PRINT7;FALLS JA, DANN BILDSCHIRM BEHANDLUNG
BB84 CDFE20 09690 CALL CP ;SONST CP AUF DRUCKER AUSGEBEN

```

```

BB87 18B8      09700      JR      PRINT1 ;UND NAECHSTE PHYSIKALISCHE ZEILE
BB89 7E        09710 PRINT7 LD      A,(HL) ;ZEICHEN FÜR LETZTE BILDSCHIRMSPALTE
BB8A CD2A03    09720      CALL     OUTCHF ;AUSGEBEN (ES ERFOLGT AUTOMATISCH CR)
BB8D 23        09730      INC      HL ;ZEIGER ZEIGT AUF NAECHSTES ZEICHEN
BB8E 18B1      09740      JR      PRINT1 ;UND NAECHSTE PHYSIKALISCHE ZEILE
BB90 CDDDBB    09750 PRINT8 CALL     PVPOS ;AKTUELLE SPALTENPOSITION NACH (IX+1)
BB93 AF        09760      XOR      A ;A=0
BB94 DDBE01    09770      CP      (IX+1) ;STEHT CURSOR AUF 1.SPALTE?
BB97 2803      09780      JR      Z,PRINT0;FALLS JA, FEIN CR AUSGEBEN
BB99 CDFE20    09790 PRINT9 CALL     CR ;SONST CR AUSGEBEN
BB9C ED4BA740 09800 PRIN10 LD      BC,(BUFANF) ;BC ZEIGT WIEDER AUF BUFFER ANFANG
BBA0 16FF      09810      LD      D,255 ;D AUF VOLLE LÄNGE SETZEN
BBA2 E1        09820      POP      HL ;HL WIEDER ALS ZEIGER ZURÜCK HOLEN
BBA3 F1        09830      POP      AF ;ZEICHEN NACH A ZURÜCK
BBA4 C9        09840      RET      ;UND RÜCKSPRUNG
09850 ;
09860 ;
09870 ;
09880 ;
09890 ;***** INDCHK
09900 ;***** (JUSTIERT INDENT-COUNTER)
09910 ;
BBA5 DD7E00    09920 INDCH1 LD      A,(IX+0);INDENT-COUNTER NACH A
BBA8 DD7704    09930      LD      (IX+4),A;A NACH AUFBEREITETEM INDENT-COUNTER
BBAB AF        09940      XOR      A ;A=0
BBAC DDB605    09950      OR      (IX+5) ;THEN-ELSE-FLAG GESETZT?
BBAD 2803      09960      JR      Z,IND15;FALLS NEIN, DANN WEITER
BBB1 DD3504    09970      DEC      (IX+4) ;SONST AUFBEREITETEN INDENT-COUNTER DEC.
09980 ;
BBB4 DDCB0846 09990 IND15 BIT      0,(IX+8);IF-INDENT-FLAG GESETZT?
BBB8 2803      10000      JR      Z,INDCH1;FALLS NEIN, DANN WEITER
BBBA DD3504    10010      DEC      (IX+4) ;SONST AUFBEREITETEN INDENT-COUNTER DEC.
10020 ;
BBBD AF        10030 INDCH1 XOR      A ;A=0
BBBE DDB604    10040      OR      (IX+4) ;IST AUFBEREITETER INDENT-COUNTER = 0?
BBC1 F2C8B5    10050      JP      P,INDCH2;FALLS JA, DANN WEITER
10060 ;
BBC4 DD360400 10070      LD      (IX+4),0;SONST AUFBEREITETEN INDENT-COUNTER = 0
10080 ;
BBC8 DD7E06    10090 INDCH2 LD      A,(IX+6);MAX. INDENT NACH A
BBCB DDBE04    10100      CP      (IX+4) ;IST INDENT KLEINER GLEICH MAX. INDENT
BBCE F2D4BB    10110      JP      P,INDCH3;FALLS JA, DANN WEITER
10120 ;
BBD1 DD7704    10130      LD      (IX+4),A;SONST INDENT GLEICH MAX. INDENT SETZEN
10140 ;
BBD4 DD360500 10150 INDCH3 LD      (IX+5),0;THEN-ELSE-FLAG RÜCK SETZEN
BBDB DD360800 10160      LD      (IX+8),0;IF-INDENT-FLAG RÜCK SETZEN
BBDC C9        10170      RET      ;UND MIT AUFBEREITETEM INDENT ZURÜCK
10180 ;
10190 ;
10200 ;
10210 ;
10220 ;***** PVPOS
10230 ;***** (BRINGT SPALTENPOSITION VON DRUCKER ODER BILDSCHIRM NACH (IX+1)
10240 ;
BBDD 3AA640    10250 PVPOS LD      A,(VPOS);VIDEO SPALTENPOSITION NACH A
BBE0 DD7701    10260      LD      (IX+1),A;WEITER NACH (IX+1)
BBE3 3A9C40    10270      LD      A,(PVFLAG) ;PVFLAG NACH A
BBE6 B7        10280      OR      A ;AKTUELLE AUSGABE AUF VIDEO?
BBE7 C8        10290      RET      Z ;FALLS JA, ZURÜCK

```

```

-----
BBE8 3A9B40 10300 LD A,(PPOS);SONST DRUCKERSPALTE NACH A
BBE9 DD7701 10310 LD (IX+1),A;UND WEITER NACH (IX+1)
BBEE C9 10320 RET ;UND ZURUECK
10330 ;
10340 ;
10350 ;
10360 ;
10370 ;***** BLANKS
10380 ;***** (GIBT 5 LEERZEICHEN AUS)
10390 ;
BBEF 0605 10400 BLANK5 LD B,5 ;SCHLEIFENZAEHLER SETZEN
BBF1 3E20 10410 BLANK LD A,' ' ;LEERZEICHEN NACH A
BBF3 CD2A03 10420 CALL OUTCH ;UND AUSGEBEN
BBF6 10F9 10430 DJNZ BLANK ;SCHLEIFE
BBF8 C9 10440 RET ;SONST ZURUECK
10450 ;
10460 ;
10470 ;
10480 ;
10490 ;***** TOF BUF
10500 ;***** (DECODIERT TOKEN UND SCHREIBT ES IN DEN BUFFER)
10510 ;
BBF9 3EFF 10520 TOF BUF LD A,255 ;A=255
BBFB BA 10530 CP D ;IST BUFFER NOCH LEER?
BBFC 280A 10540 JR Z,TOF BU1;FALLS JA, WEITER
BBFE 3E20 10550 LD A,' ' ;LEERZEICHEN NACH A
BC00 DDBE02 10560 CP (IX+2) ;IST LETZTES AUSGEGEBENES Z. LEERZEICHEN?
BC03 2803 10570 JR Z,TOF BU1;FALLS JA, WEITER
BC05 CD2DBB 10580 CALL INBUF ;SONST LEERZEICHEN IN DEN BUFFER
BC08 7E 10590 TOF BU1 LD A,(HL) ;ATUELLES TOKEN NACH A
BC09 DD360700 10600 LD (IX+7),0;DOPPELTOKEN-FLAG RUCH SETZEN
BC0D FEFF 10610 CP DOPTOK ;IST ATUELLES TOKEN EIN DOPPELTOKEN?
BC0F 2004 10620 JR NZ,NODOTO ;FALLS NEIN, FLAG RUCHGESETZT LASSEN
BC11 DD360701 10630 LD (IX+7),1;SONST DOPPELTOKEN-FLAG SETZEN
BC15 23 10640 NODOTO INC HL ;ZEIGER FUER ROM-EINSPRUNG JUSTIEREN
BC16 D67F 10650 SUB 7FH ;TOKEN - 7F LIEFERT NR DES KEYWORD
BC18 E5 10660 PUSH HL ;CODEZEIGER RETTEN
BC19 5F 10670 LD E,A ;NR. DES KEYWORD NACH E
BC1A CDAD39 10680 CALL TABANF ;LAEDT DEN ANFANG DER KEYWORDTAB. NACH HL
BC1D 7E 10690 TOF BU2 LD A,(HL) ;ZEICHEN LADEN
BC1E B7 10700 OR A ;ANFANG DES NAECHSTEN KEYWORD ERREICHT?
BC1F 23 10710 INC HL ;ZEIGER AUF NAECHSTES ZEICHEN
BC20 F21DBC 10720 JP P,TOF BU2;FALLS NEIN, DANN WEITERSUCHEN
BC23 1D 10730 DEC E ;GEWUNSCHTES KEYWORD ERREICHT?
BC24 20F7 10740 JR NZ,TOF BU2 ;FALLS NEIN, DANN WEITERSUCHEN
BC26 E67F 10750 AND 7FH ;WORTANFANGSMARKIERUNG ENTFERNEN
BC28 CD2DBB 10760 TOF BU3 CALL INBUF ;UND ZEICHEN IN DEN BUFFER
BC2B 7E 10770 LD A,(HL) ;ZEICHEN DES KEYWORD LADEN
BC2C 23 10780 INC HL ;ZEIGER AUF NAECHSTES ZEICHEN
BC2D B7 10790 OR A ;1. ZEICHEN DES NAECHSTEN KEYWORD?
BC2E F228BC 10800 JP P,TOF BU3;FALLS NEIN, DANN WEITER AUSGEBEN
BC31 E1 10810 POP HL ;CODEZEIGER ZURUECK
BC32 2B 10820 DEC HL ;UND ZURUECK JUSTIEREN
BC33 AF 10830 XOR A ;A=0
BC34 DDBE07 10840 CP (IX+7) ;DOPPELTOKENFLAG RUCHGESETZT?
BC37 2810 10850 JR Z,TOF BU4;FALLS JA, NORMALER AUSSPRUNG
BC39 7E 10860 LD A,(HL) ;BEARBEITETES TOKEN NACH A
BC3A FE81 10870 CP FCOLOUR ;WAR ES DAS FCOLOUR-TOKEN?
BC3C 200B 10880 JR NZ,TOF BU4 ;FALLS NEIN, NORMALER AUSSPRUNG
BC3E 23 10890 INC HL ;ZEIGER AUF ZEICHEN NACH TOKEN SETZEN

```

```

BC3F 7E      10900      LD      A,(HL) ;UND ZEICHEN NACH A
BC40 2B      10910      DEC     HL ;ZEIGER FÜR AUSSPRUNG JUSTIEREN
BC41 FE52    10920      CP      'R' ;IST ES 'R'?
BC43 2004    10930      JR      NZ,TOFBU4 ;FALLS NEIN, NORMALER AUSSPRUNG
BC45 CD2DBB  10940      CALL    INBUF ;'R' IN DEN BUFFER LADEN
BC48 23      10950      INC     HL ;ZEIGER ZUM AUSSPRUNG JUSTIEREN
BC49 3E20    10960      LD      A,' ' ;LEERZEICHEN NACH A
BC4B CD2DBB  10970      CALL    INBUF ;UND AUSGEBEN
BC4E C9      10980      RET

10990 ;
11000 ;
11010 ;
11020 ;
11030 ;***** VARIABLENSPEICHER
11040 ;
BC4F 00      11050      VARADR  DEFB    0 ;(IX+0) INDENT COUNTER
BC50 00      11060      DEFB    0 ;(IX+1) PRINT BZW. VIDEO-SPALTE
BC51 00      11070      DEFB    0 ;(IX+2) LETZTES IN BUFFER GESCHR. ZEICHEN
BC52 00      11080      DEFB    0 ;(IX+3) VORGEGEBENE ZEILENLAENGE
BC53 00      11090      DEFB    0 ;(IX+4) AUFBEREITETER INDENT COUNTER
BC54 00      11100      DEFB    0 ;(IX+5) THEN-ELSE-FLAG (1=GESETZT)
BC55 00      11110      DEFB    0 ;(IX+6) MAX. INDENT VORGABE
BC56 00      11120      DEFB    0 ;(IX+7) DOPPELTOFEN-FLAG (1=DOPPELTOFEN)
BC57 00      11130      DEFB    0 ;(IX+8) IF-INDENT-FLAG
BC58 00      11140      DEFB    0 ;(IX+9) IF-FLAG
BC59 00      11150      DEFB    0 ;(IX+10) ELSE-FLAG
11160 ;
11170 ;***** SUCHEN IM BASIC-TEXT FÜR TFS-80 U. COLOUR-GENIE *****
11180 ;***** VERSION 1.0 28.11.83 *****
11190 ;***** (C) 1983 BY M. BUNING *****
11200 ;
11210 ;***** ROM-ADRESSEN
11220 ;
032A      11230      SHOCHR  EQU     32AH ;GIBT ZEICHEN IN A AUS
0361      11240      GETSTR  EQU     361H ;LIEST ZEILE IN DEN BUFFER
0FAF      11250      SHOLIN  EQU     0FAFH ;GIBT ZEILENNUMMER IN HL AUS
1BC0      11260      CODE    EQU     1BC0H ;ERZEUGT ZWISCHENCODE
2B75      11270      SHMSG   EQU     2B75H ;GIBT STRING AUS
1D9B      11280      STOP    EQU     1D9BH ;WERTET BREAK UND SHIFT FL. AUS
11290 ;
11300 ;***** SYSTEM-RAM-ADRESSEN
11310 ;
40A4      11320      BSTF    EQU     40A4H ;ZEIGER AUF BASIC-ANFANG
40A6      11330      CURSP   EQU     40A6H ;CURSOR-SPALTE
11340 ;
11350 ;***** TOF EN
11360 ;
0088      11370      DATAT   EQU     88H ;DATA
0093      11380      PENT    EQU     93H ;PEM
11390 ;
11400 ;
11410 ;
11420 ;
11430 ;START DER SUCH-ROUTINE
11440 ;
11450 ;
11460 ;
11470 ;
11480 ;
BC5A DD21D5B0 11490      GET    LD      IX,FLAG ;IX ALS ZEIGER AUF DEN FLAG-BEREICH

```

```

BC5E DD360000 11500      LD      (IX+0),0      ;T-FLAG RÜCKSETZEN
BC62 3E01      11510      LD      A,1
BC64 32D4BD    11520      LD      (NF),A
BC67 2192BD    11530      LD      HL,STRM
BC6A CD752B    11540      CALL    SHOMSG
BC6D CD6103    11550 STRIN  CALL    GETSTR
BC70 23        11560      INC     HL
BC71 7E        11570      LD      A,(HL)
BC72 B7        11580      OR      A
BC73 CA16BD    11590      JP      Z,RETURN      ;RÜCKKEHR, FALLS NICHTS EINGEGEBEN
BC76 FE40      11600      CP      'S'
BC78 200F      11610      JR      NZ,ST2
BC7A 23        11620      INC     HL
BC7B 7E        11630      LD      A,(HL)
BC7C FE40      11640      CP      'S'
BC7E 2008      11650      JR      NZ,ST1
BC80 23        11660      INC     HL
BC81 7E        11670      LD      A,(HL)
BC82 B7        11680      OR      A
BC83 CA16BD    11690      JP      Z,RETURN      ;RÜCKKEHR, FALLS NICHTS EINGEGEBEN
BC86 1809      11700      JR      ST3
BC88 2B        11710 ST1   DEC     HL
BC89 CDC01B    11720 ST2   CALL    CODE
BC8C DD360001 11730      LD      (IX+0),1      ;T-FLAG SETZEN
BC90 23        11740      INC     HL
BC91 22D0BD    11750 ST3   LD      (STR),HL
BC94 2AA440    11760      LD      HL,(BSP)
BC97 7E        11770      LD      A,(HL)
BC98 B7        11780      OR      A
BC99 2006      11790      JR      NZ,SU
BC9B 23        11800      INC     HL
BC9C 7E        11810      LD      A,(HL)
BC9D B7        11820      OR      A
BC9E 2860      11830      JR      Z,NOPGM
BCA0 2B        11840      DEC     HL
BCA1 2B        11850 SU    DEC     HL
BCA2 23        11860 GETPTR INC     HL
BCA3 5E        11870      LD      E,(HL)
BCA4 23        11880      INC     HL
BCA5 56        11890      LD      D,(HL)
BCA6 AF        11900      XOR     A
BCA7 B3        11910      OR      E
BCA8 C2AFBC    11920      JP      NZ,GETLIN
BCAB B2        11930      OR      D
BCAC CA09BD    11940      JP      Z,ENDBAS
BCAF 23        11950 GETLIN INC     HL
BCB0 5E        11960      LD      E,(HL)
BCB1 23        11970      INC     HL
BCB2 56        11980      LD      D,(HL)
BCB3 ED53D2BD 11990      LD      (LIN),DE
BCB7 23        12000 SCAN  INC     HL
BCB8 0600      12010      LD      B,0
BCBA ED5BD0BD 12020      LD      DE,(STR)
BCBE 1A        12030 SC1   LD      A,(DE)
BCBF B7        12040      OR      A
BCC0 C2C8BC    12050      JP      NZ,SC3
BCC3 CD6EBD    12060      CALL    PRLINE      ;AKTUELLE ZEILENNUMMER AUSGEBEN
BCC6 1830      12070      JR      SC4
BCC8 4E        12080 SC3   LD      C,(HL)
BCC9 0C        12090      INC     C

```

```

BCCA 0D      12100      DEC      C
BCCB CAA2BC  12110      JP      Z,GETPTR
BCCD B9      12120      CP      C
BCCF DD360100 12130      LD      (IX+1),0;U-FLAG RUCH SETZEN
BCD3 2004    12140      JR      NZ,SC31;BEI NICHT-ÜBEREINSTIMMUNG FLAG LASSEN
BCD5 DD360101 12150      LD      (IX+1),1;SONST U-FLAG SETZEN
BCD9 AF      12160 SC31    XOR      A      ;A=0
BCDA DDBE00  12170      CP      (IX+0) ;IST T-FLAG RUCHGESETZT?
BCDD 280D    12180      JR      Z,SC32 ;FALLS JA, DANN SPRUNG
BCDF 79      12190      LD      A,C ;SONST TEXTZEICHEN NACH A
BCE0 FE93    12200      CP      REMT ;IST ES REM-TOKEN?
BCE2 2836    12210      JR      Z,SREM ;FALLS JA, DANN AUSWERTEN
BCE4 FE88    12220      CP      DATAT ;IST ES DATA-TOKEN?
BCE6 283E    12230      JR      Z,SDATA ;FALLS JA, DANN AUSWERTEN
BCE8 FE22    12240      CP      "" ;IST ES ANFUHRUNGSZEICHEN?
BCEA 2861    12250      JR      Z,SQUOTE;FALLS JA, AUSWERTEN
BCEC AF      12260 SC32    XOR      A      ;A=0
BCED DDBE01  12270      CP      (IX+1) ;U-FLAG GESETZT?
BCF0 2806    12280      JR      Z,SC4 ;FALLS NEIN, DANN NEUER SUCHBEGINN
BCF2 04      12290 SC35    INC      B
BCF3 23      12300      INC      HL
BCF4 13      12310      INC      DE
BCF5 C3BEBC  12320      JP      SC1
BCF8 4B      12330 SC4     LD      C,B
BCF9 AF      12340      XOR      A
BCFA 47      12350      LD      B,A
BCFB ED42    12360      SBC      HL,BC
BCFD C3B7BC  12370      JP      SCAN
BD00 21B0BD  12380 NOPGM   LD      HL,NPGM
BD03 CD752B  12390      CALL     SHOMSG
BD06 C316BD  12400      JP      RETURN
BD09 3AD4BD  12410 ENDBAS  LD      A,(NF)
BD0C B7      12420      OR      A
BD0D CA16BD  12430      JP      Z,RETURN
BD10 21A1BD  12440      LD      HL,NFM
BD13 CD752B  12450      CALL     SHOMSG
BD16 21D7BD  12460 RETURN  LD      HL,NULL ;ZEIGER AUF 3 NULLEN
BD19 C9      12470      RET
          12480 ;
BD1A CD60BD  12490 SREM    CALL     EQUAL
BD1D 23      12500 SREM1   INC      HL ;HL ZEIGT AUF NAECHSTES TEXTZEICHEN
BD1E 7E      12510      LD      A,(HL) ;UND ZEICHEN NACH A
BD1F FE00    12520      CP      0 ;ZEILENENDE ERREICHT?
BD21 20FA    12530      JR      NZ,SREM1 ;FALLS NEIN, DANN SCHLEIFE
BD23 C3A2BC  12540      JP      GETPTR ;SONST ZUR NAECHSTEN ZEILE
          12550 ;
BD26 CD60BD  12560 SDATA   CALL     EQUAL
BD29 23      12570 SDATA1  INC      HL ;HL AUF NAECHSTES ZEICHEN
BD2A 7E      12580      LD      A,(HL) ;U. ZEICHEN NACH A
BD2B FE00    12590      CP      0 ;ZEILENENDE ERREICHT?
BD2D 2003    12600      JR      NZ,SDATA2 ;FALLS NEIN, WEITER
BD2F C3A2BC  12610      JP      GETPTR ;SONST NEUE ZEILE
BD32 FE22    12620 SDATA2  CP      "" ;QUOTE?
BD34 200F    12630      JR      NZ,SDATA3 ;FALLS NEIN, WEITERE ABFRAGE
BD36 23      12640 SDATA3  INC      HL ;ZEIGER AUF NAECHSTES ZEICHEN
BD37 7E      12650      LD      A,(HL) ;U. NACH A
BD38 FE00    12660      CP      0 ;ZEILENENDE ERREICHT?
BD3A 2003    12670      JR      NZ,SDATA4 ;FALLS NEIN, DANN WEITER
BD3C C3A2BC  12680      JP      GETPTR ;SONST NEUE ZEILE
BD3F FE22    12690 SDATA4  CP      "" ;QUOTE?

```

```

BD41 20F3      12700      JR      NZ,SDATA3      ;FALLS NEIN, DANN SCHLEIFE
BD43 18E4      12710      JR      SDATA1          ;SONST HAUPTSCHLEIFE
BD45 FE3A      12720 SDATA5 CP      ':'          ;NAECHSTES STATEMENT?
BD47 20E0      12730      JR      NZ,SDATA1      ;FALLS NEIN, HAUPTSCHLEIFE
BD49 2B        12740      DEC     HL              ;SONST IN ZEILE WEITERSUCHEN
BD4A C3B7BC    12750      JP      SCAN
                12760 ;
BD4D CD60BD    12770 SQUOTE CALL     EQUAL
BD50 23        12780 SQUOT1 INC      HL              ;HL AUF NAECHSTES ZEICHEN
BD51 7E        12790      LD      A,(HL)          ;U. NACH A
BD52 FE00      12800      CP      0              ;ZEILENENDE ERREICHT?
BD54 2003      12810      JR      NZ,SQUOT2      ;FALLS NICHT, DANN WEITER
BD56 C3A2BC    12820      JP      GETPTR          ;SONST NEUE ZEILE
BD59 FE22      12830 SQUOT2 CP      '"'          ;ERNEUT QUOTE?
BD5B 20F3      12840      JR      NZ,SQUOT1      ;FALL NEIN, SCHLEIFE
BD5D C3B7BC    12850      JP      SCAN            ;SONST IN DIESER ZEILE WEITERSUCHEN
                12860 ;
                12870 ;
                12880 ;
                12890 ;
                12900 ;***** BENUTZTE UNTERPROGRAMME
                12910 ;
BD60 AF        12920 EQUAL  XOR      A              ;A=0
BD61 DDBE01    12930      CP      (IX+1)          ;IST ÜBEREINSTIMMUNGSFLAG GESETZT?
BD64 C8        12940      RET      Z              ;FALLS NEIN, RÜCKSPRUNG
BD65 13        12950      INC     DE              ;DE AUF NAECHSTES ZEICHEN IN STRING
BD66 1A        12960      LD      A,(DE)          ;UND NACH A
BD67 FE00      12970      CP      0              ;IST STRINGENDE ERREICHT?
BD69 C0        12980      RET      NZ              ;FALLS NICHT, DANN RÜCKSPRUNG
BD6A CD6EBD    12990      CALL    PRLINE          ;SONST ZEILENUMMER AUSGEBEN
BD6D C9        13000      RET
                13010 ;
                13020 ;
                13030 ;
                13040 ;
BD6E E5        13050 PRLINE PUSH     HL
BD6F C5        13060      PUSH    BC
BD70 CD9B1D    13070      CALL    STOP            ;STOPPEN WAHREND DES AUSGEBEN
BD73 AF        13080      XOR      A
BD74 32D4BD    13090      LD      (NF),A
BD77 3AA640    13100      LD      A,(CURSP)
BD7A FE21      13110      CP      33             ;57 BEI TRS-80, 33 BEI COLOUR-GENIE
BD7C FAB4BD    13120      JP      M,PRLIN1
BD7F 3E0D      13130      LD      A,13
BD81 CD2A03    13140      CALL    SHOCHR
BD84 2AD2BD    13150 PRLIN1 LD      HL,(LIN)
BD87 CDAF0F    13160      CALL    SHOLIN
BD8A 3E20      13170      LD      A,20H
BD8C CD2A03    13180      CALL    SHOCHR
BD8F C1        13190      POP      BC
BD90 E1        13200      POP      HL
BD91 C9        13210      RET
                13220 ;
                13230 ;
                13240 ;
                13250 ;
                13260 ;***** TEXTE
                13270 ;
BD92 1D1E      13280 STRM   DEFW     1E1DH
BD94 53        13290      DEFM     'Suchen nach:'

```



```

-----
BDA0 00      13300      DEFB      0
BDA1 4E      13310 NFM      DEFM      'Nicht gefunden'
BDAF 00      13320      DEFB      0
BDB0 4B      13330 NPGM      DEFM      'Kein BASIC-Programm im Speicher'
BDCF 00      13340      DEFB      0
          13350 ;
          13360 ;
          13370 ;
          13380 ;
          13390 :***** VARIABLEN-BEREICH
          13400 ;
BDD0 0000    13410 STR      DEFW      0      ;ZEIGER AUF STRINGANFANG
BDD2 0000    13420 LIN      DEFW      0      ;ZWISCHENSPEICHER FÜR ZEILENUMMER
BDD4 00      13430 NF      DEFB      0      ;FLAG (1=KEINE ÜBEREINSTIMM. GEFUNDEN)
BDD5 00      13440 FLAG      DEFB      0      ;FLAG (1=TOFFEN-MODE)
BDD6 00      13450      DEFB      0      ;FLAG (1=BYTE-ÜBEREINSTIMMUNG)
BDD7 00      13460 NULL      DEFB      0
BDD8 00      13470      DEFB      0
BDD9 00      13480      DEFB      0
          13490      ; * BASIC-PROGRAMM-PACIER *
          13500      ; ENTFERNT ALLE LEERRÄUME UND KOMMENTARE
          13510      ; COPYRIGHT BY LUDIGER ROEDERATH, 26.9.81
          13520      ;
          13530      ; BASICUNTERPROGRAMME
          13540      ;
1D78      13550 CHRGET      EQU      1D78H      ;RST 10H UP
1F07      13560 SEOL      EQU      1F07H      ;ZEILENENDE SUCHEN
1B10      13570 LARG      EQU      1B10H      ;2 ARGUMENTE ANALYSIEREN
1E5B      13580 DECLN1      EQU      1E5BH      ;ZNR DECODIEREN
1B2C      13590 SLINE      EQU      1B2CH      ;ZEILE IM PROGRAMMTEXT SUCHEN
0FA7      13600 PRZNR1      EQU      0FA7H      ;' IN ' ZNR AUSGEBEN
0FAF      13610 PRZNR      EQU      0FAFH      ;ZNR AUSGEBEN
2B75      13620 PRINTT      EQU      2B75H      ;TEXT AUSGEBEN
20FE      13630 CF      EQU      20FEH      ;CR AUSGEBEN
1AE8      13640 ENDML      EQU      1AE8H      ;HAUPTSCHLEIFENENDE
1E4A      13650 FCEFF      EQU      1E4AH      ;FC-ERROR
          13660      ;
          13670      ; BASICZEIGER
          13680      ;
1929      13690 TEXT6      EQU      1929H      ;TEXT 'READY'
40A4      13700 TEXTA      EQU      40A4H      ;ZEIGER AUF PROGRAMMTEXTANFANG
40F9      13710 VARA      EQU      40F9H      ;ZEIGER AUF ANFANG DER VARTAB
4191      13720 FILLV      EQU      4191H
          13730      ;
          13740      ; TOFFENS
          13750      ;
008D      13760 GOTO      EQU      8DH
008E      13770 RUN      EQU      8EH
0091      13780 GOSUB      EQU      91H
0093      13790 REM      EQU      93H
0095      13800 ELSE      EQU      95H
009F      13810 RESUME      EQU      9FH
00CA      13820 THEN      EQU      0CAH
0098      13830 DATA      EQU      88H
00AA      13840 FILLT      EQU      0AAH
          13850      ;
          13860      ; FILL
          13870      ;
BDDA 280F    13880 FILL      JR      Z,FILL2
BDDC 280D    13890      JF      C,FILL2

```

| | | | |
|-------------|--------------|------|------------|
| BDDE FECE | 13900 | CP | OCEH |
| BDE0 2809 | 13910 | JR | Z,KILL2 |
| BDE2 FE2E | 13920 | CP | " " |
| BDE4 2805 | 13930 | JR | Z,KILL2 |
| BDE6 2B | 13940 | DEC | HL |
| BDE7 D7 | 13950 | RST | 10H |
| BDE8 C33B01 | 13960 | JP | SNERR |
| BDEB CD101B | 13970 KILL2 | CALL | LARG |
| BDEE D1 | 13980 | POP | DE |
| BDEF D5 | 13990 | PUSH | DE |
| BDF0 C5 | 14000 | PUSH | BC |
| BDF1 CD2C1B | 14010 | CALL | SLINE |
| BDF4 69 | 14020 | LD | L,C |
| BDF5 60 | 14030 | LD | H,B |
| BDF6 D1 | 14040 | POP | DE |
| BDF7 DF | 14050 | RST | 18H |
| BDF8 D5 | 14060 | PUSH | DE |
| BDF9 DA4A1E | 14070 | JP | C,FCERR |
| BDFC CD74BF | 14080 | CALL | LEN |
| BDFE 2191BF | 14090 | LD | HL,TEXT1 |
| BE02 CD752B | 14100 | CALL | PRINTT |
| BE05 2AA440 | 14110 | LD | HL,(TEXTA) |
| BE08 2B | 14120 | DEC | HL |
| BE09 D5 | 14130 | PUSH | DE |
| BE0A D1 | 14140 REF1 | POP | DE |
| BE0B 23 | 14150 | INC | HL |
| BE0C 7E | 14160 | LD | A,(HL) |
| BE0D 23 | 14170 | INC | HL |
| BE0E B6 | 14180 | OR | (HL) |
| BE0F 287D | 14190 | JR | Z,PACK |
| BE11 23 | 14200 | INC | HL |
| BE12 5E | 14210 | LD | E,(HL) |
| BE13 23 | 14220 | INC | HL |
| BE14 56 | 14230 | LD | D,(HL) |
| BE15 D5 | 14240 | PUSH | DE |
| BE16 D7 | 14250 REF2 | RST | 10H |
| BE17 A7 | 14260 | AND | A |
| BE18 28F0 | 14270 | JR | Z,REF1 |
| BE1A FE22 | 14280 | CP | " " |
| BE1C 200B | 14290 | JR | NZ,JUMP |
| BE1E D7 | 14300 STRING | RST | 10H |
| BE1F A7 | 14310 | AND | A |
| BE20 28E8 | 14320 | JR | Z,REF1 |
| BE22 FE22 | 14330 | CP | " " |
| BE24 20F8 | 14340 | JR | NZ,STRING |
| BE26 FEFF | 14350 JUMP | CP | OFFH |
| BE28 23 | 14360 | INC | HL |
| BE29 28EB | 14370 | JR | Z,REF2 |
| BE2B 2B | 14380 | DEC | HL |
| BE2C FE8D | 14390 | CP | GOTO |
| BE2E 2819 | 14400 | JR | Z,JUMP1 |
| BE30 FE91 | 14410 | CP | GOSUB |
| BE32 2815 | 14420 | JR | Z,JUMP1 |
| BE34 FE8E | 14430 | CP | RUN |
| BE36 2811 | 14440 | JR | Z,JUMP1 |
| BE38 FE9F | 14450 | CP | RESUME |
| BE3A 280D | 14460 | JR | Z,JUMP1 |
| BE3C FE95 | 14470 | CP | ELSE |
| BE3E 2804 | 14480 | JR | Z,JUMP2 |
| BE40 FECA | 14490 | CP | THEN |

| | | | |
|-------------|--------------|------|------------|
| BE42 20D2 | 14500 | JR | NZ, REF2 |
| BE44 D7 | 14510 JUMP2 | RST | 10H |
| BE45 2B | 14520 | DEC | HL |
| BE46 30CE | 14530 | JR | NC, REF2 |
| BE48 AF | 14540 | XOR | A |
| BE49 F5 | 14550 JUMP1 | PUSH | AF |
| BE4A CD5B1E | 14560 | CALL | DECLN1 |
| BE4D 2B | 14570 | DEC | HL |
| BE4E F1 | 14580 | POP | AF |
| BE4F D69F | 14590 | SUB | RESUME |
| BE51 B2 | 14600 | OR | D |
| BE52 B3 | 14610 | OR | E |
| BE53 28C1 | 14620 | JR | Z, REF2 |
| BE55 E5 | 14630 | PUSH | HL |
| BE56 CD2C1B | 14640 | CALL | SLINE |
| BE59 301C | 14650 | JR | NC, UNDEFL |
| BE5B 69 | 14660 | LD | L, C |
| BE5C 60 | 14670 | LD | H, B |
| BE5D 23 | 14680 | INC | HL |
| BE5E 23 | 14690 | INC | HL |
| BE5F 23 | 14700 | INC | HL |
| BE60 D7 | 14710 | RST | 10H |
| BE61 FE3A | 14720 | CP | ?, ? |
| BE63 CC781D | 14730 | CALL | Z, CHRGET |
| BE66 FE93 | 14740 | CP | REM |
| BE6B 2002 | 14750 | JR | NZ, REF3 |
| BE6A 36FE | 14760 | LD | (HL), OFEH |
| BE6C E1 | 14770 REF3 | POP | HL |
| BE6D D7 | 14780 | RST | 10H |
| BE6E FE2C | 14790 | CP | ?, ? |
| BE70 3E8D | 14800 | LD | A, GOTO |
| BE72 28D5 | 14810 | JR | Z, JUMP1 |
| BE74 2B | 14820 | DEC | HL |
| BE75 189F | 14830 | JR | REF2 |
| BE77 21C6BF | 14840 UNDEFL | LD | HL, TEXT3 |
| BE7A CD752B | 14850 | CALL | PRINTT |
| BE7D EB | 14860 | EX | DE, HL |
| BE7E CDAF0F | 14870 | CALL | PRZNR |
| BE81 D1 | 14880 | POP | DE |
| BE82 E1 | 14890 | POP | HL |
| BE83 E5 | 14900 | PUSH | HL |
| BE84 D5 | 14910 | PUSH | DE |
| BE85 CDA70F | 14920 | CALL | PRZNR1 |
| BE88 E1 | 14930 | POP | HL |
| BE89 CDFE20 | 14940 | CALL | CR |
| BE8C 1888 | 14950 | JR | REF2 |
| BE8E 21B5BF | 14960 PACK | LD | HL, TEXT2 |
| BE91 CD752B | 14970 | CALL | PRINTT |
| BE94 E1 | 14980 | POP | HL |
| BE95 D1 | 14990 | POP | DE |
| BE96 E5 | 15000 | PUSH | HL |
| BE97 D5 | 15010 | PUSH | DE |
| BE98 2B | 15020 | DEC | HL |
| BE99 23 | 15030 PACKL | INC | HL |
| BE9A 7E | 15040 PACKL1 | LD | A, (HL) |
| BE9B 23 | 15050 | INC | HL |
| BE9C B6 | 15060 | OR | (HL) |
| BE9D CA3ABF | 15070 | JP | Z, PACKE |
| BEA0 23 | 15080 | INC | HL |
| BEA1 5E | 15090 | LD | E, (HL) |

| | | | |
|-------------|--------------|------|-----------|
| BEA2 23 | 15100 | INC | HL |
| BEA3 56 | 15110 | LD | D, (HL) |
| BEA4 23 | 15120 | INC | HL |
| BEA5 E3 | 15130 | EY | (SP), HL |
| BEA6 DF | 15140 | RST | 10H |
| BEA7 E7 | 15150 | EX | (SP), HL |
| BEA8 DABF | 15160 | JP | C, PAGE |
| BEA9 E5 | 15170 | PUSH | HL |
| BEAC 7E | 15180 CHAR | LD | A, (HL) |
| BEAD A7 | 15190 | AND | A |
| BEAE 2878 | 15200 | JP | Z, EOL |
| BEF0 FEFF | 15210 | CP | OFFH |
| BEB2 23 | 15220 | INC | HL |
| BEB3 23 | 15230 | INC | HL |
| BEB4 28F6 | 15240 | JR | Z, CHAR |
| BEB6 2F | 15250 | DEC | HL |
| BEB7 2B | 15260 | DEC | HL |
| BEBS FE20 | 15270 | CP | " " |
| BEBA 2824 | 15280 | JR | Z, SPACE |
| BEFC FE0A | 15290 | CP | 0AH |
| BEBE 2820 | 15300 | JR | Z, SPACE |
| BEC0 FE93 | 15310 | CP | REM |
| BEC2 282B | 15320 | JR | Z, DELREM |
| BEC4 FEFE | 15330 | CP | OFFH |
| BEC6 2826 | 15340 | JR | Z, DELR1 |
| BEC8 FE88 | 15350 | CP | DATA |
| BECA 287A | 15360 | JR | Z, FDATA |
| BEC0 FE3A | 15370 | CP | ": " |
| BECE 2829 | 15380 | JR | Z, COLON |
| BED0 FE22 | 15390 CPSTR | CP | " " |
| BED2 23 | 15400 CHAR1 | INC | HL |
| BED3 20D7 | 15410 | JR | NZ, CHAR |
| BED5 2B | 15420 | DEC | HL |
| BED6 D7 | 15430 STR1 | RST | 10H |
| BED7 A7 | 15440 | AND | A |
| BED8 284E | 15450 | JR | Z, EOL |
| BEDA D622 | 15460 | SUB | " " |
| BEDC 20F8 | 15470 | JR | NZ, STR1 |
| BEDE 18F0 | 15480 | JR | CPSTR |
| BEE0 CDE5BE | 15490 SPACE | CALL | USPACE |
| BEE3 18C7 | 15500 | JR | CHAR |
| BEE5 E5 | 15510 USPACE | PUSH | HL |
| BEE6 D7 | 15520 | RST | 10H |
| BEE7 D1 | 15530 | POP | DE |
| BEE8 D5 | 15540 DEL | PUSH | DE |
| BEE9 CD61BF | 15550 | CALL | DELETE |
| BEEC E1 | 15560 | POP | HL |
| BEED C9 | 15570 | RET | |
| BEEE 23 | 15580 DELR1 | INC | HL |
| BEEF E5 | 15590 DELREM | PUSH | HL |
| BEF0 CD071F | 15600 | CALL | SEOL |
| BEF3 D1 | 15610 | POP | DE |
| BEF4 CDE8BE | 15620 | CALL | DEL |
| BEF7 18B3 | 15630 | JR | CHAR |
| BEF9 E5 | 15640 COLON | PUSH | HL |
| BEFA D7 | 15650 | RST | 10H |
| BEFB E1 | 15660 | POP | HL |
| BEFC FE93 | 15670 | CP | REM |
| BEFE 28E0 | 15680 | JR | Z, SPACE |
| BF00 FEFE | 15690 | CP | OFFH |

| | | | | |
|------|--------|-------|--------|-------------|
| BF02 | 28DC | 15700 | JR | Z,SPACE |
| BF04 | 18CC | 15710 | JR | CHAR1 |
| BF06 | 23 | 15720 | PDATA | INC |
| BF07 | 7E | 15730 | LD | A, (HL) |
| BF08 | FE20 | 15740 | CP | ' ' |
| BF0A | CCE5BE | 15750 | CALL | Z,USPACE |
| BF0D | 2B | 15760 | DEC | HL |
| BF0E | D7 | 15770 | RST | 10H |
| BF0F | 289B | 15780 | JR | Z,CHAR |
| BF11 | FE22 | 15790 | CP | ' ' |
| BF13 | 200D | 15800 | JR | NZ,SC11 |
| BF15 | D7 | 15810 | DS | RST |
| BF16 | A7 | 15820 | AND | A |
| BF17 | 280F | 15830 | JR | Z,EOL |
| BF19 | FE22 | 15840 | CP | ' ' |
| BF1B | 20F8 | 15850 | JR | NZ,DS |
| BF1D | 18E7 | 15860 | JR | PDATA |
| BF1F | D7 | 15870 | SC | RST |
| BF20 | 288A | 15880 | JR | Z,CHAR |
| BF22 | FE2C | 15890 | SC11 | CP |
| BF24 | 20F9 | 15900 | JR | NZ,SC |
| BF26 | 18DE | 15910 | JR | PDATA |
| BF28 | D1 | 15920 | EOL | POP |
| BF29 | DF | 15930 | RST | 18H |
| BF2A | C299BE | 15940 | JP | NZ,PACKL |
| BF2D | 1B | 15950 | DEC | DE |
| BF2E | 1B | 15960 | DEC | DE |
| BF2F | 1B | 15970 | DEC | DE |
| BF30 | 1B | 15980 | DEC | DE |
| BF31 | D5 | 15990 | PUSH | DE |
| BF32 | 23 | 16000 | INC | HL |
| BF33 | CD61BF | 16010 | CALL | DELETE |
| BF36 | E1 | 16020 | POP | HL |
| BF37 | C39ABE | 16030 | JP | PACKL1 |
| BF3A | D1 | 16040 | PACKE | POP |
| BF3B | 2AA440 | 16050 | LD | HL, (TEXTA) |
| BF3E | 2B | 16060 | DEC | HL |
| BF3F | 23 | 16070 | DELREF | INC |
| BF40 | 7E | 16080 | LD | A, (HL) |
| BF41 | 23 | 16090 | INC | HL |
| BF42 | B6 | 16100 | OR | (HL) |
| BF43 | CA55BF | 16110 | JP | Z,END |
| BF46 | 23 | 16120 | INC | HL |
| BF47 | 23 | 16130 | INC | HL |
| BF48 | 23 | 16140 | INC | HL |
| BF49 | 7E | 16150 | LD | A, (HL) |
| BF4A | FEFE | 16160 | CP | OFEH |
| BF4C | 2002 | 16170 | JR | NZ,NEXTL1 |
| BF4E | 3693 | 16180 | LD | (HL),REM |
| BF50 | CD071F | 16190 | NEXTL1 | CALL |
| BF53 | 18EA | 16200 | JR | DELREF |
| BF55 | CD74BF | 16210 | END | CALL |
| BF58 | 212919 | 16220 | LD | HL,TEXT6 |
| BF5B | CD752B | 16230 | CALL | PRINTT |
| BF5E | C3E81A | 16240 | JP | ENDML |
| BF61 | D5 | 16250 | DELETE | PUSH |
| BF62 | EB | 16260 | EX | DE,HL |
| BF63 | 2AF940 | 16270 | LD | HL, (VARA) |
| BF66 | A7 | 16280 | AND | A |
| BF67 | ED52 | 16290 | SBC | HL,DE |

```

BF69 4D      16300      LD      C,L
BF6A 44      16310      LD      B,H
BF6B EB      16320      EX      DE,HL
BF6C D1      16330      POP     DE
BF6D EDB0    16340      LDIR
BF6F ED53F940 16350      LD      (VARA),DE
BF73 C9      16360      RET
BF74 E5      16370      PUSH    HL
BF75 21E4BF  16380      LD      HL,TEXT4
BF78 CD752B  16390      CALL    PRINTT
BF7B 2AF940  16400      LD      HL,(VARA)
BF7E ED5BA440 16410      LD      DE,(TEXTA)
BF82 37      16420      SCF
BF83 ED52    16430      SBC     HL,DE
BF85 2B      16440      DEC     HL
BF86 CDADF0F 16450      CALL    PRZNR
BF89 21F8BF  16460      LD      HL,TEXT5
BF8C CD752B  16470      CALL    PRINTT
BF8F E1      16480      POP     HL
BF90 C9      16490      RET
BF91 5A      16500      TEXT1   DEFM    'Zeilenreferenzen werden ermittelt.'
BFB3 0D      16510      DEFB     0DH
BFB4 00      16520      DEFB     0
BFB5 50      16530      TEXT2   DEFM    'Packen beginnt.'
BFC4 0D      16540      DEFB     0DH
BFC5 00      16550      DEFB     0
BFC6 42      16560      TEXT3   DEFM    'Bezug auf undefinierte Zeile '
BFE3 00      16570      DEFB     0
BFE4 50      16580      TEXT4   DEFM    'Programm erfordert '
BFF7 00      16590      DEFB     0
BFFB 20      16600      TEXT5   DEFM    ' Bytes'
BFFE 0D      16610      DEFB     0DH
BFFF 00      16620      DEFB     0
B67C        16630      END      INIT
00000 TOTAL ERRORS
01403 TEXT AREA BYTES LEFT

```

2.3.1 Feldaufzeichnung ARRAY

=====

ARRAY ermöglicht das schnelle Aufzeichnen und Laden von Feldern auf Kassette oder Diskette durch einen einzigen Befehl, wobei für das ganze Feld nur ein Header auf Kassette aufgezeichnet wird.

Funktionsweise:

ARRAY schreibt numerische Felder wie sie im Speicher stehen auf Kassette bzw. Diskette. Bei Stringfeldern wird jeder String - angeführt von seiner Länge - in der Reihenfolge, in der die Strings im Feld angeordnet sind, aufgezeichnet. Die Betriebssystemanbindung ist ähnlich wie bei PACK.

Der Aufruf:

** Kassettenversion:

ARRAY stellt zwei neue Befehle zur Verfügung, die folgendermaßen aufgerufen werden:

FIELD PRINT #-1,A(20),B\$(20),C(10,10),...

FIELD INPUT #-1,A(20),B\$(20),C(10,10),...

Durch den FIELD PRINT-Befehl werden die Feldelemente A(0)..A(20), B\$(0..20), C(0,0)..C(10,10) auf Kassette aufgezeichnet. Die Variablenliste kann beliebig lang sein. Bei eindimensionalen ARRAY kann auch nur ein Teil aufgezeichnet werden:

```
10 DIM A(20)
```

```
.
```

```
.
```

```
100 FIELD PRINT #-1,A(10)
```

Hier werden nur die ersten 11 Elemente aufgezeichnet.

Bei mehrdimensionalen Feldern ist es zwar auch möglich nur einen Teil aufzuzeichnen, aber da immer der Bereich im Speicher zwischen Feld(0,0,...,0) und Feld(X1,X2,...,XN) aufgezeichnet wird, kann im allgemeinen nicht davon ausgegangen werden, daß die Elemente Feld(0..X1,0..X2,...,0..XN) darunter sind.

Mit der FIELD INPUT-Anweisung müssen genausoviele Elemente eingelesen werden, wie aufgezeichnet wurden, ansonsten erfolgt ein BS-Error.

Sollen mit einem FIELD-INPUT Stringfelder gelesen werden, so werden zuerst diese Felder gelöscht und dann der Stringspace von überflüssigen Strings bereinigt um für die zu lesenden Strings genügend Platz zu schaffen. Da dies

unter Umständen längere Zeit in Anspruch nehmen kann, wird bei Beginn des eigentlichen Ladevorgangs ein 'S' in der oberen rechten Bildschirmecke angezeigt. Sollte sich später beim Laden herausstellen, daß der Platz nicht ausreicht, wird das Laden mit einem OS-Error abgebrochen.

Beim Laden wird bei jeder Null bzw. bei jedem neuen String der Stern in der oberen rechten Ecke umgeschaltet.

**** Diskettenversion:**

ARRAY stellt zwei neue Befehle zur Verfügung, die folgendermaßen aufgerufen werden:

FIELD PRINT #1,A(20),B\$(20),C(10,10),...

FIELD INPUT #1,A(20),B\$(20),C(10,10),...

Durch den FIELD PRINT-Befehl werden die Feldelemente A(0)..A(20), B\$(0..20), C(0,0)..C(10,10) auf Diskette (File 1) aufgezeichnet. Die Variablenliste kann beliebig lang sein. Bei eindimensionalen ARRAY kann auch nur ein Teil aufgezeichnet werden:

```
10 DIM A(20)
```

```
.
```

```
.
```

```
100 FIELD PRINT #1,A(10)
```

Hier werden nur die ersten 11 Elemente aufgezeichnet. Bei mehrdimensionalen Feldern ist es zwar auch möglich nur einen Teil aufzuzeichnen, aber da immer der Bereich im Speicher zwischen Feld(0,0,...,0) und Feld(X1,X2,...,XN) aufgezeichnet wird, kann im allgemeinen nicht davon ausgegangen werden, daß die Elemente Feld(0..X1,0..X2,...,0..XN) darunter sind.

Mit der FIELD INPUT-Anweisung müssen genausoviele Elemente eingelesen werden, wie aufgezeichnet wurden, ansonsten erfolgt ein BS-Error. Reicht bei Zeichenkettenfeldern der Platz im Stringspace nicht aus, wird der Ladevorgang mit OS-Error abgebrochen. Ist ein Garbage-Collect notwendig, kann es vorkommen, daß das Laufwerk für diese Zeit anhält.

2.3.2 Felddarstellung ARRAY - Quellprogramm, Kass Seite 121

```

00100 ;ASEG
BE6F 00110 ORG 0BE6FH
BE6F 2A7D41 00120 INIT LD HL, (FIELDV)
BE72 228ABE 00130 LD (AAZ+1), HL
BE75 217EBE 00140 LD HL, AAY
BE78 227D41 00150 LD (FIELDV), HL
BE7B C3191A 00160 JP 1A19H
00170 ;
BE7E FE89 00180 AAY CP INPUT
BE80 CA0DBF 00190 JP Z, ABI
BE83 FEB2 00200 CP PRINT
BE85 2805 00210 JR Z, ABA
BE87 2B 00220 DEC HL
BE88 D7 00230 RST 10H
BE89 C30000 00240 AAZ JP 0
BE8C D7 00250 ABA RST 10H
BE9D CDA935 00260 CALL INITW
BE9D 2B 00270 DEC HL
BE91 010000 00280 ABB LD BC, 0
BE94 CD6000 00290 CALL DELAY
BE97 CD6000 00300 CALL DELAY
BE9A D7 00310 RST 10H
BE9B CDD0BF 00320 CALL ABF
BE9E 7B 00330 LD A, E
BE9F CD1F02 00340 CALL WRITEB
BEA2 7A 00350 LD A, D
BEA3 CD1F02 00360 CALL WRITEB
BEA6 3AAF40 00370 LD A, (TYPX)
BEA9 FE03 00380 CP 3
BEAB 2049 00390 JR NZ, ABG
BEAD E5 00400 PUSH HL
BEAE C5 00410 PUSH BC
BEAF D5 00420 PUSH DE
BEB0 210000 00430 LD HL, 0
BEB3 0A 00440 ABC LD A, (BC)
BEB4 85 00450 ADD A, L
BEB5 6F 00460 LD L, A
BEB6 7C 00470 LD A, H
BEB7 CE00 00480 ADC A, 0
BEB9 67 00490 LD H, A
BEBA 1B 00500 DEC DE
BEBB 1F 00510 DEC DE
BEBC 1B 00520 DEC DE
BEBD 03 00530 INC BC
BEBE 03 00540 INC BC
BEBF 03 00550 INC BC
BEC0 7A 00560 LD A, D
BEC1 B0 00570 OR E
BEC2 20EF 00580 JR NZ, ABC
BEC4 7D 00590 LD A, L
BEC5 CD1F02 00600 CALL WRITEB
BEC8 7C 00610 LD A, H
BEC9 CD1F02 00620 CALL WRITEB
BECC D1 00630 POP DE
BECD E1 00640 POP HL
BECE D5 00650 ABD PUSH DE
BECF 56 00660 LD D, (HL)
BED0 23 00670 JNC HL
BED1 4E 00680 LD C, (HL)
BED2 23 00690 INC HL

```

2.3.2 Felddarstellung ARRAY - Quellprogramm, Kass Seite 122

```

BED3 45      00700      LD      E, (HL)
BED4 23      00710      INC     HL
BED5 7A      00720      LD      A, D
BED6 CD1F02  00730      CALL   WRITER
BED7 A7      00740      AND     A
BED8 280E    00750      JP      Z, ABF
BEDC 0A      00760 ABE   LD      A, (BC)
BEDD CD1F02  00770      CALL   WRITER
BEE0 CDCABF  00780      CALL   BREAK
BEE3 C2CABF  00790      JP      NZ, RET2
BEE6 03      00800      INC     BC
BEE7 15      00810      DEC     D
BEE8 20F2    00820      JR      NZ, ABE
BEEA D1      00830 ABF   POP     DE
BEEB 1B      00840      DEC     DE
BEEC 1B      00850      DEC     DE
BEDD 1B      00860      DEC     DE
BEEE 7A      00870      LD      A, D
BEEF B7      00880      OR      E
BEF0 20DC    00890      JR      NZ, ABD
BEF2 E1      00900      POP     HL
BEF3 C306BF  00910      JP      ABH
BEF6 0A      00920 ABG   LD      A, (BC)
BEF7 CD1F02  00930      CALL   WRITER
BEFA CDCABF  00940      CALL   BREAK
BEFD C1C6BF  00950      JP      NZ, RET
BF00 03      00960      INC     BC
BF01 1B      00970      DEC     DE
BF02 7A      00980      LD      A, D
BF03 B7      00990      OR      E
BF04 20F0    01000      JR      NZ, ABG
BF06 7E      01010 ABH   LD      A, (HL)
BF07 FE2C    01020      CP      ', '
BF09 CA91BE  01030      JP      Z, ABB
BF0C C9      01040      RET
BF0D CD2828  01050 ABI   CALL   2828H
BF10 D7      01060      RST     10H
BF11 CDB535  01070      CALL   35B5H
BF14 2B      01080      DEC     HL
BF15 E5      01090      PUSH   HL
BF16 AF      01100      XOR     A
BF17 32FFBF  01110      LD      (FLAG), A
BF1A D7      01120 ABJ1  RST     10H
BF1B CDD0BF  01130      CALL   ABF
BF1E 3AAF40  01140      LD      A, (TYPX)
BF21 FE03    01150      CP      3
BF23 200B    01160      JR      NZ, ABF1
BF25 32FFBF  01170      LD      (FLAG), A
BF28 AF      01180 AB1 1  XOR     A
BF29 02      01190      LD      (BC), A
BF2A 03      01200      INC     BC
BF2B 1B      01210      DEC     DE
BF2C 7A      01220      LD      A, D
BF2D B7      01230      OR      E
BF2E 20F8    01240      JP      NZ, AB1 1
BF30 7E      01250 ABF1  LD      A, (HL)
BF31 FE2C    01260      CP      ', '
BF33 CA1ABF  01270      JP      Z, ABJ1
BF36 3AFFBF  01280      LD      A, (FLAG)
BF39 A7      01290      AND     A

```

2.3.2 Felddarstellung ARRAY - Quellprogramm, Kass Seite 123

| | | | |
|-------------|-----------|------|-------------|
| BF3A C4E629 | 01300 | CALL | NZ, COLLEC |
| BF3D 3E53 | 01310 | LD | A, '5' |
| BF3F 322744 | 01320 | LD | (4427H), A |
| BF42 CD4C02 | 01330 | CALL | INTR |
| BF45 E1 | 01340 | POP | HL |
| BF46 D7 | 01350 ABJ | RST | 10H |
| BF47 CDD0BF | 01360 | CALL | ABP |
| BF4A CDED01 | 01370 | CALL | READB |
| BF4D BB | 01380 | CP | E |
| BF4E C23D27 | 01390 | JP | NZ, PSERP |
| BF51 CDED01 | 01400 | CALL | READB |
| BF54 BA | 01410 | CP | D |
| BF55 C23D27 | 01420 | JP | NZ, PSERP |
| BF58 3AAF40 | 01430 | LD | A, (T/EX) |
| BF5B FE07 | 01440 | CP | 3 |
| BF5D 204A | 01450 | JR | NZ, ABP |
| BF5F E5 | 01460 | PUSH | HL |
| BF60 CDED01 | 01470 | CALL | READB |
| BF63 6F | 01480 | LD | L, A |
| BF64 CDED01 | 01490 | CALL | READB |
| BF67 67 | 01500 | LD | H, H |
| BF68 CDE8BF | 01510 | CALL | ABP |
| BF6B D2FABF | 01520 | JP | NZ, ABT |
| BF6E 60 | 01530 | LD | L, C |
| BF6F 60 | 01540 | LD | H, B |
| BF70 CDE401 | 01550 ABM | CALL | BLINK |
| BF73 D5 | 01560 | PUSH | DE |
| BF74 CDED01 | 01570 | CALL | READB |
| BF77 77 | 01580 | LD | (HL), A |
| BF78 23 | 01590 | INC | HL |
| BF79 E5 | 01600 | PUSH | HL |
| BF7A 2AD640 | 01610 | LD | HL, (STR25) |
| BF7D 4F | 01620 | LD | C, A |
| BF7E 0600 | 01630 | LD | B, 0 |
| BF80 A7 | 01640 | AND | A |
| BF81 ED42 | 01650 | SBC | HL, BC |
| BF83 22D640 | 01660 | LD | (STR26), HL |
| BF86 27 | 01670 | INC | HL |
| BF87 EB | 01680 | EX | DE, HL |
| BF88 E1 | 01690 | POP | HL |
| BF89 73 | 01700 | LI | (HL), E |
| BF8A 23 | 01710 | INC | HL |
| BF8B 72 | 01720 | LD | (HL), D |
| BF8C 23 | 01730 | INC | HL |
| BF8D A7 | 01740 | AND | A |
| BF8E 280E | 01750 | JP | Z, AB0 |
| BF90 CDED01 | 01760 ABN | CALL | READB |
| BF93 12 | 01770 | LD | (DE), A |
| BF94 CDCABF | 01780 | CALL | BREAR |
| BF97 C2C4BF | 01790 | JP | NZ, RET2 |
| BF9A 13 | 01800 | INC | DE |
| BF9B 0D | 01810 | DEC | C |
| BF9C 20F2 | 01820 | JR | NZ, ABN |
| BF9E D1 | 01830 ABO | POP | DE |
| BF9F 1B | 01840 | DEC | DE |
| BFA0 1B | 01850 | DEC | DE |
| BFA1 1B | 01860 | DEC | DE |
| BFA2 7B | 01870 | LD | A, E |
| BFA3 B2 | 01880 | OR | D |
| BFA4 20CA | 01890 | JR | NZ, ABM |

2.3.2 Felddarstellung ARRAY - Quellprogramm, Kass Seite 124

| | | | |
|---------------|--------------|------|--------------|
| BFA6 E1 | 01900 | POP | HL |
| BFA7 1814 | 01910 | JR | ABQ |
| BFA9 CDED01 | 01920 ABP | CALL | READB |
| BFAC 02 | 01930 | LD | (BC), A |
| BFAD A7 | 01940 | AND | A |
| BFAE CCE401 | 01950 | CALL | Z, BLINK |
| BFB1 CDCABF | 01960 | CALL | BREAK |
| BFB4 C2C6BF | 01970 | JP | NZ, RET |
| BFB7 03 | 01980 | INC | BC |
| BFB8 1B | 01990 | DEC | DE |
| BFB9 7A | 02000 | LD | A, D |
| BFBA B3 | 02010 | OR | E |
| BFB8 20EC | 02020 | JR | NZ, ABP |
| BFBD 7E | 02030 ABQ | LD | A, (HL) |
| BFBE FE2C | 02040 | CP | ' , ' |
| BFC0 CA46BF | 02050 | JP | Z, ABJ |
| BFC3 C9 | 02060 | RET | |
| BFC4 D1 | 02070 RET2 | POP | DE |
| BFC5 E1 | 02080 | POP | HL |
| BFC6 2AE640 | 02090 RET | LD | HL, (40E6H) |
| BFC9 C9 | 02100 | RET | |
| BFCA 3A40F8 | 02110 BREAK | LD | A, (0F840H) |
| BFCD CB57 | 02120 | BIT | 2, A |
| BFCF C9 | 02130 | RET | |
| BFD0 CD0D26 | 02140 ABR | CALL | SVAR |
| BFD3 E5 | 02150 | PUSH | HL |
| BFD4 2AFB40 | 02160 | LD | HL, (ARRTAB) |
| BFD7 ED52 | 02170 | SBC | HL, DE |
| BFD9 D2F60A | 02180 | JP | NC, TMERR |
| BFDC 3AAF40 | 02190 | LD | A, (TYPX) |
| BFDF 6F | 02200 | LD | L, A |
| BFE0 2600 | 02210 | LD | H, 0 |
| BFE2 19 | 02220 | ADD | HL, DE |
| BFE3 ED42 | 02230 | SBC | HL, BC |
| BFE5 EB | 02240 | EX | DE, HL |
| BFE6 E1 | 02250 | POP | HL |
| BFE7 C9 | 02260 | RET | |
| BFE8 D5 | 02270 ABS | PUSH | DE |
| BFE9 E5 | 02280 | PUSH | HL |
| BFEA 2AD640 | 02290 | LD | HL, (STRZ6) |
| BFED A7 | 02300 | AND | A |
| BFEE ED5BA040 | 02310 | LD | DE, (STRSFC) |
| BFF2 ED52 | 02320 | SBC | HL, DE |
| BFF4 EB | 02330 | EX | DE, HL |
| BFF5 13 | 02340 | INC | DE |
| BFF6 E1 | 02350 | POP | HL |
| BFF7 DF | 02360 | RST | 18H |
| BFF8 D1 | 02370 | POP | DE |
| BFF9 C9 | 02380 | RET | |
| BFFA 1E1A | 02390 ABT | LD | E, OSERR |
| BFFC C3A219 | 02400 | JP | ERROR |
| BFFF 00 | 02410 FLAG | DEFB | 0 |
| 001A | 02420 OSERR | EQU | 001AH |
| 0060 | 02430 DELAY | EQU | 0060H |
| 0089 | 02440 INPUT | EQU | 0089H |
| 00B2 | 02450 PRINT | EQU | 00B2H |
| 01E4 | 02460 BLINK | EQU | 01E4H |
| 01ED | 02470 READB | EQU | 01EDH |
| 021F | 02480 WRITEB | EQU | 021FH |
| 35A9 | 02490 INITW | EQU | 35A9H |

2.3.2 Feldaufzeichnung ARRAY - Quellprogramm, Kass Seite 125

```
-----  
024C      02500 INTR  EQU  024CH  
0AF6      02510 TMERR EQU  0AF6H  
19A2      02520 ERROR EQU  19A2H  
260D      02530 SVAR  EQU  260DH  
273D      02540 BSERR EQU  273DH  
28E6      02550 COLLEC EQU  28E6H  
40A0      02560 STRSPC EQU  40A0H  
40AF      02570 TYPY  EQU  40AFH  
40D6      02580 STRZG  EQU  40D6H  
40FB      02590 ARRTAB EQU  40FBH  
417D      02600 FIELDV EQU  417DH  
BE6F      02610      END  INIT  
00000 TOTAL ERRORS  
31585 TEXT AREA BYTES LEFT
```

2.3.3 Felddarstellung ARRAY - Quellprogramm, Disk Seite 126

```

                                00100      ;ASEG
BE9D      00110      ORG      0BE9DH
                                00120      ;
BE9D 2A7D41 00130 INIT      LD      HL, (FIELDV)
BEA0 22B8BE 00140      LD      (AAZ+1), HL
BEA3 21ACBE 00150      LD      HL, AAY
BEA6 227D41 00160      LD      (FIELDV), HL
BEA9 C3191A 00170      JP      1A19H
                                00180      ;
41CA      00190 WDISK      EQU      41CAH
41BE      00200 WDISKE      EQU      41BEH
032A      00210 OUTCH      EQU      032AH
0013      00220 INCHW      EQU      0013H
C7BD      00230 BUFFER      EQU      0C7BDH
                                00240      ;
BEAC FE89      00250 AAY      CP      INPUT
BEAE CA28EF 00260      JP      Z, ABI
BEB1 FEB2      00270      CP      PRINT
BEB3 2805      00280      JR      Z, ABA
BEB5 2B      00290      DEC      HL
BEB6 D7      00300      RST      10H
BEB7 C30000 00310 AAZ      JP      0
BEBA D7      00320 ABA      RST      10H
BEBB CDCA41 00330      CALL      WDISK
BEBE 2B      00340      DEC      HL
BEBF D7      00350 ABB      RST      10H
BEC0 CDBBBF 00360      CALL      SARRAY
BEC3 7B      00370      LD      A, E
BEC4 CD2A03 00380      CALL      OUTCH
BEC7 7A      00390      LD      A, D
BEC8 CD2A03 00400      CALL      OUTCH
BECB 3AAF40 00410      LD      A, (TYPX)
BECE FE03      00420      CP      3
BED0 2043      00430      JR      NZ, ABG
BED2 E5      00440      PUSH      HL
BED3 C5      00450      PUSH      BC
BED4 D5      00460      PUSH      DE
BED5 210000 00470      LD      HL, 0
BED8 0A      00480 ABC      LD      A, (BC)
BED9 85      00490      ADD      A, L
BEDA 6F      00500      LD      L, A
BEDB 7C      00510      LD      A, H
BEDC CE00      00520      ADC      A, 0
BEDE 67      00530      LD      H, A
BEDF 1B      00540      DEC      DE
BEE0 1B      00550      DEC      DE
BEE1 1B      00560      DEC      DE
BEE2 03      00570      INC      BC
BEE3 03      00580      INC      BC
BEE4 03      00590      INC      BC
BEE5 7A      00600      LD      A, D
BEE6 B3      00610      OR      E
BEE7 20EF      00620      JR      NZ, ABC
BEE9 7D      00630      LD      A, L
BEEA CD2A03 00640      CALL      OUTCH
BEED 7C      00650      LD      A, H
BEEE CD2A03 00660      CALL      OUTCH
BEF1 D1      00670      POP      DE
BEF2 E1      00680      POP      HL
BEF3 D5      00690 ABD      PUSH      DE

```

2.3.3 Felddarstellung ARRAY - Quellprogramm, Disk Seite 127

```

BEF4 56      00700      LD      D, (HL)
BEF5 23      00710      INC     HL
BEF6 4E      00720      LD      C, (HL)
BEF7 23      00730      INC     HL
BEF8 46      00740      LD      B, (HL)
BEF9 23      00750      INC     HL
BEFA 7A      00760      LD      A, D
BEFB CD2A03  00770      CALL    OUTCH
BEFC A7      00780      AND     A
BEFD 2908    00790      JR      Z, ABF
BEFE 0A      00800 ABE    LD      A, (BC)
BEFF CD2A03  00810      CALL    OUTCH
BF00 03      00820      INC     BC
BF01 15      00830      DEC     D
BF02 20F8    00840      JR      NZ, ABE
BF03 D1      00850 ABF    POP     DE
BF04 1B      00860      DEC     DE
BF05 1F      00870      DEC     DE
BF06 1B      00880      DEC     DE
BF07 7A      00890      LD      A, D
BF08 B7      00900      OR      E
BF09 20E2    00910      JR      NZ, ABD
BF0A E1      00920      POP     HL
BF0B C31FBF  00930      JP      ABH
BF0C 0A      00940 ABG    LD      A, (BC)
BF0D CD2A03  00950      CALL    OUTCH
BF0E 03      00960      INC     BC
BF0F 1F      00970      DEC     DE
BF10 7A      00980      LD      A, D
BF11 B7      00990      OR      E
BF12 20F6    01000      JR      NZ, AEG
BF13 7E      01010 ABH    LD      A, (HL)
BF14 FE2C    01020      CP      ', '
BF15 CAFEFE  01030      JP      Z, ABF
BF16 C7BE41  01040      JP      WDISE
BF17 CD2828  01050 ABJ    CALL    Z8L8H
BF18 CDBDC7  01060      CALL    BUFFER
BF19 CF      01070      RST     8H
BF1A 2C      01080      DEFB   ', '
BF1B E5      01090      PUSH    HL
BF1C 210200  01100      LD      HL, 2
BF1D 09      01110      ADD     HL, BC
BF1E 22FEBF  01120      LD      (FCB), HL
BF1F F1      01130      POP     HL
BF20 2B      01140      DEC     HL
BF21 D7      01150 ABJ    RST     10H
BF22 CDBBBF  01160      CALL    SARRAY
BF23 CDEABF  01170      CALL    READP
BF24 BB      01180      CP      E
BF25 C23D27  01190      JP      NZ, 3SERP
BF26 CDEABF  01200      CALL    READP
BF27 BA      01210      CP      D
BF28 C23D27  01220      JP      NZ, BEERR
BF29 7A8F40  01230      LD      A, (TYPX)
BF2A FE03    01240      CP      3
BF2B 2057    01250      JP      NZ, ABF
BF2C EF      01260      PUSH    HL
BF2D CDEABF  01270      CALL    READP
BF2E 6F      01280      LD      L, A
BF2F CDEABF  01290      CALL    READP

```

2.3.3 Felddarstellung ARRAY - Quellprogramm, Disk Seite 128

| | | | |
|-------------|-----------|------|------------|
| BF5B 67 | 01300 | LD | H,A |
| BF5C CDD3BF | 01310 | CALL | ABS |
| BF5F 3817 | 01320 | JR | C,ABL |
| BF61 C5 | 01330 | PUSH | BC |
| BF62 D5 | 01340 | PUSH | DE |
| BF63 AF | 01350 ABK | XOR | A |
| BF64 02 | 01360 | LD | (BC),A |
| BF65 03 | 01370 | INC | BC |
| BF66 1B | 01380 | DEC | DE |
| BF67 7A | 01390 | LD | A,D |
| BF68 B3 | 01400 | OR | E |
| BF69 20F8 | 01410 | JR | NZ,ABK |
| BF6B E5 | 01420 | PUSH | HL |
| BF6C CDE628 | 01430 | CALL | COLLEC |
| BF6F E1 | 01440 | POP | HL |
| BF70 D1 | 01450 | POP | DE |
| BF71 CDD3BF | 01460 | CALL | ABS |
| BF74 C1 | 01470 | POP | BC |
| BF75 D2E5BF | 01480 | JP | NC,ABT |
| BF78 69 | 01490 AEL | LD | L,C |
| BF79 60 | 01500 | LD | H,B |
| BF7A D5 | 01510 ABM | PUSH | DE |
| BF7B CDEABF | 01520 | CALL | READB |
| BF7E 77 | 01530 | LD | (HL),A |
| BF7F 23 | 01540 | INC | HL |
| BF80 E5 | 01550 | PUSH | HL |
| BF81 2AD640 | 01560 | LD | HL,(STRZG) |
| BF84 4F | 01570 | LD | C,A |
| BF85 0600 | 01580 | LD | B,0 |
| BF87 A7 | 01590 | AND | A |
| BF88 ED42 | 01600 | SBC | HL,BC |
| BF8A 22D640 | 01610 | LD | (STRZG),HL |
| BF8D 23 | 01620 | INC | HL |
| BF8E EB | 01630 | EX | DE,HL |
| BF8F E1 | 01640 | POP | HL |
| BF90 73 | 01650 | LD | (HL),E |
| BF91 23 | 01660 | INC | HL |
| BF92 72 | 01670 | LD | (HL),D |
| BF93 23 | 01680 | INC | HL |
| BF94 A7 | 01690 | AND | A |
| BF95 2808 | 01700 | JR | Z,ABO |
| BF97 CDEABF | 01710 ABN | CALL | READB |
| BF9A 12 | 01720 | LD | (DE),A |
| BF9B 13 | 01730 | INC | DE |
| BF9C 0D | 01740 | DEC | C |
| BF9D 20F8 | 01750 | JR | NZ,ABN |
| BF9F D1 | 01760 ABO | POP | DE |
| BFA0 1B | 01770 | DEC | DE |
| BFA1 1B | 01780 | DEC | DE |
| BFA2 1B | 01790 | DEC | DE |
| BFA3 7B | 01800 | LD | A,E |
| BFA4 B2 | 01810 | OR | D |
| BFA5 20D3 | 01820 | JR | NZ,ABM |
| BFA7 E1 | 01830 | POP | HL |
| BFA8 180A | 01840 | JR | ABO |
| BFAA CDEABF | 01850 ABP | CALL | READB |
| BFAD 02 | 01860 | LD | (BC),A |
| BFAE 03 | 01870 | INC | BC |
| BFAF 1B | 01880 | DEC | DE |
| BFB0 7A | 01890 | LD | A,D |

2.3.3 Felddarstellung ARRAY - Quellprogramm, Disk Seite 129

```

BFB1 B3      01900      OR      E
BFB2 20F5    01910      JP      NZ,ABF
BFB4 7E      01920 ABD    LD      A,(HL)
BFB5 FE2C    01930      CP      ', '
BFB7 C43ABF  01940      JP      Z,ABJ
BFBA C9      01950      RET
BFBB CD0D26  01960 SARRAY CALL    SVAR
BFBE E5      01970      PUSH    HL
BFBF 2AFB40  01980      LD      HL,(ARPTAB)
BFC2 ED52    01990      SBC     HL,DE
BFC4 D2F60A  02000      JP      NC,TMERR
BFC7 2AAF40  02010      LD      A,(TYPX)
BFCA 6F      02020      LD      L,A
BFCE 2600    02030      LD      H,0
BFCD 19      02040      ADD     HL,DE
BFCE ED42    02050      SBC     HL,BC
BFD0 EB      02060      EX      DE,HL
BFD1 E1      02070      POP     HL
BFD2 C9      02080      RET
BFD3 D5      02090 ABS     PUSH    DE
BFD4 E5      02100      PUSH    HL
BFD5 2AD640  02110      LD      HL,(STRZG)
BFD8 A7      02120      AND     A
BFD9 ED5BA040 02130      LD      DE,(STRSPC)
BFDD ED52    02140      SBC     HL,DE
BFDF EB      02150      EX      DE,HL
BFE0 13      02160      INC     DE
BFE1 E1      02170      POP     HL
BFE2 DF      02180      RST     18H
BFE3 D1      02190      POP     DE
BFE4 C9      02200      RET
BFE5 1E1A    02210 ABT     LD      E,DSERR
BFE7 C3A219  02220      JP      ERROR
BFEA D5      02230 READB   PUSH    DE
BFEB ED5BFEBF 02240      LD      DE,(FCB)
BFEF CD1300  02250      CALL    INCHW
BFF2 D1      02260      POP     DE
BFF3 C8      02270      RET     Z
BFF4 CBFF    02280      SET     7,A
BFF5 CD0944  02290      CALL    4409H
BFF9 1E66    02300      LD      E,102
BFFB C3A219  02310      JP      ERROR
BFFE 0000    02320 FCB     DEFW    0
                02330      ;
001A          02340 DSERR   EQU     001AH
0089          02350 INPUT   EQU     0089H
00B2          02360 PRINT   EQU     00B2H
0AF5          02370 TMERR   EQU     0AF5H
19A2          02380 ERROR   EQU     19A2H
260D          02390 SVAR    EQU     260DH
273D          02400 BSERR   EQU     273DH
28E6          02410 COLLEC  EQU     28E6H
40A0          02420 STRSPC  EQU     40A0H
40AF          02430 TYPX    EQU     40AFH
40D6          02440 STRZ5   EQU     40D6H
40FB          02450 ARPTAB  EQU     40FBH
417D          02460 FIELDV  EQU     417DH+1
BE9D          02470      END     INIT
000000 TOTAL ERRORS
71943 TEXT AREA BYTES LEFT

```

2.4.1 Kopierprogramm COPBIT

=====

COPBIT kopiert Colour-Genie-Kassetten Bit für Bit und ermöglicht somit einfaches Kopieren praktisch aller Colour-Genie-Kassetten ohne Kenntnis des Aufzeichnungsformates:

- 1) Systemkassetten
- 2) Basickassetten
- 3) Datenkassetten
- 4) geschützte Formate
z.B. mit Preloader und/oder Baudrateumschaltung

Funktion

COPBIT liest Kassetten Bit für Bit ein und achtet dabei nicht auf das Format. Unterbrechungen mit Bit-Strom werden erkannt und führen auf Wunsch automatisch zum Abbruch des Ladens. Erst nach dem Laden wird das Format analysiert und bei Systemfiles werden die Prüfsummen, bei Basicfiles das Pointersystem überprüft. Beim Schreiben auf Kassette wird der eingelesene Bit-Strom (auf eine ganze Anzahl von Bytes ergänzt) aus dem Speicher wieder auf Kassette kopiert.

Preloader, Baudrateumschaltung und Autostart stellen für COPBIT keine Probleme dar.

COPBIT wurde mit dem MACRO-80 von Microsoft entwickelt. Dabei wurden viele seiner besonderen Möglichkeiten ausgenutzt. Insbesondere sind dies die MACRO's und die Pseudo-Anweisungen .PHASE und .DEPHASE. Nur so konnte das Programm derartig gestaltet werden, daß der vorhandene Speicher optimal ausgenutzt wird (siehe Technische Anmerkung). Diese Version ist natürlich nicht für den COLASM geeignet. Daher ist das Programm in einer für den COLASM bearbeiteten Version aufgelistet, die auch auf der Quellprogramm-kassette- und diskette enthalten ist. Diese Version läßt etwas weniger Speicher als unten angegeben für die zu kopierenden Programme frei.

Technische Anmerkung:

COPBIT benutzt verschiedene Speicherbereiche, die im Colour-Genie-System eigentlich für bestimmte Systemaufgaben reserviert sind. Daher ist es nicht möglich nach dem Laden von COPBIT einen RESET durchzuführen. Dies führt unweigerlich zu einer Fehlfunktion, da dadurch viele Programmteile von COPBIT zerstört werden. Ein Teil von COPBIT ist im Speicher für die freiprogrammierbaren Zeichen untergebracht, wodurch manchmal kleine Bildschirmstörungen auftreten. Durch diese intensive Ausnutzung des Speicher war es aber möglich 30526 (=773EH im 32K-System, im 16K-System: 14142 = 373EH) Bytes für das zu kopierende File zur Verfügung zu stellen.

Hauptmenue

Nachdem COPBIT erfolgreich geladen ist, meldet sich das Hauptmenue:

COPBIT V3.0 Kassettenkopierprogramm

Kopiert beliebige Kassetten bitweise.

(C) 1984 Luidger Roeckrath, 5100 Aachen

```

+-----+-----+-----+
I Funktionen:           I A I      I
+-----+-----+-----+
I                                     I
I <L> a d e n                I
I                                     I
I <U> e r g l e i c h e n    I
I                                     I
I <S> p e i c h e r n        I
I                                     I
I <F> e h l e r t a p e l a d e n I
I                                     I
I <A> u t o m a t i k (E i n / A u s) I
I                                     I
I <D> r u c k e r (E i n / A u s)  I
I                                     I
+-----+-----+-----+

```

Durch Antippen der Buchstaben in <> kann die entsprechende Funktion angewählt werden. Ist die jeweilige Funktion beendet, wird automatisch ins Hauptmenue zurückgekehrt.

In den beiden Kästen hinter "Funktion:" wird der Zustand der Flags für die Automatik und den Drucker angezeigt. "A" bedeutet Automatik an, "D" bedeutet Drucker ein.

Laden

Nach der Aufforderung, den Kassettenrecorder wiedergabebereit zu machen und deren Bestätigung, beginnt COPBIT das Programm zu lesen. Sobald der Header erkannt wurde, erscheinen in der oberen, rechten Bildschirmcke die zwei Sterne. Das File wird von da an Byte für Byte eingelesen und im Speicher abgelegt. Sollte eine Aufzeichnung am Ende kein vollständiges Byte mehr haben (bei geschützten Formaten), werden auch die restlichen Bits richtig geladen und gespeichert.

Während des Ladens läuft in der Bildschirmmitte der eingelesene Text durch. Weiterhin wird bei Systemfiles ein F gefolgt vom Filenamen, ein B für jeden Block und ein E für den Entrypointblock angezeigt.

Beispiel: COLMON

FCOLMONBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBE

Das Laden wird unter folgenden Bedingungen gestoppt:

1) Automatik Ein (Startbedingung)

sobald eine Pause in der Aufzeichnung auftritt.

2) Automatik Aus

sobald eine Pause auftritt und die <BREAK>-Taste gedrückt wird.

Will man den Ladevorgang mitten in einer Aufzeichnung beenden, kann man die Pause durch Anhalten des Kassettenrecorder erzeugen. Auch wenn der Header gesucht wird, also bevor die zwei Sterne erscheinen, ist es möglich, den Ladevorgang abubrechen. Hier ist nur erforderlich die <BREAK>-Taste zu drücken, der Kassettenrekorder braucht nicht angehalten zu werden.

Analyse der geladenen Aufzeichnung

Nach Beendigung des Ladevorgangs wird die geladenen Aufzeichnung analysiert. Dabei wird festgestellt, ob es sich um ein System- oder Basicfile handelt:

1) Systemfile:

Das Format für Systemfiles ist folgendes (siehe auch Colour-Genie-ROM-Listing):

Filename: 55H 6 Bytes Filename
 Block: 3CH Länge Startadresse (LSB/MSB) Datenbytes
 Prüfsumme
 Entry: 78H Entrypoint (LSB/MSB)

Beispiel: COLMON (6000H-6FFFH)

Filename: 55H 43H 4FH 4CH 4DH 4FH 4EH = 55H COLMON
 Block : 3CH 80H 00H 60H (128 Datenbytes) (Prüfsumme)
 = 6000H-607FH
 <30 ähnliche Blöcke>
 Block : 3CH 80H 80H 6FH (128 Datenbytes) (Prüfsumme)
 = 6F80H-6FFFH
 Entry : 78H 00H 60H = 6000H

COPBIT erkennt automatisch das Systemformat und gibt in obigen Beispiel aus:

Systemfile: COLMON
 Block : 6000H-6FFFH
 Entry : 6000H

Bilden mehrere Blöcke des Systemfiles einen zusammenhängenden Speicherblock, wird nur dieser angezeigt (siehe Beispiel.).

Bei Blöcken zwischen 1 und 3 Bytes Länge wird zusätzlich der Inhalt des Blocks angezeigt. Da diese Blöcke oft Sprungvektoren sind, die überschrieben werden, ist diese Information besonders wertvoll. Ein Sprungbefehl der auf die Adresse 41E2H-41E4H geladen wird, bedeutet einen Autostart. Sie können also sofort die Startadresse eines Programmes mit Autostart ablesen, z.B.:

Block : 41E2H-41E4H : C3H E8H 41H

bedeutet, daß sobald der Entry im Systemfile erreicht, automatisch die Adresse 41E8H angesprungen wird.

Bei fehlerhaften Blöcken wird nur der Block des Systemfiles, der den Fehler enthielt, entsprechend markiert ausgedruckt:

Systemfile: COLMON
 Block : 6000H-637FH
 Block : 6380H-63FFH!!! Fehler !!!

Block : 6400H-6FFFH
Entry : 6000H

Vor jedem Block (55H, 3CH, 78H) können zusätzliche Bytes stehen, die vom System-Loader im ROM einfach überlesen werden. COPBIT zeigt deren Anzahl an. Bei Fehler erscheint auch häufig diese Anzeige:

Systemfile: COLMON
Block : 6000H-637FH
Block : 6380H-63FFH!!! Fehler !!!
zus. Bytes: 0073H
Block : 6480H-64FFH
Entry : 6000H

Auch nach dem Entry können bei geschützten Formaten zusätzliche Bytes auftreten.

2) Basicprogramme

Basicprogramme werden folgendermaßen aufgezeichnet:

Filename: 1 Byte
Basictext, wie im Speicher einschließlich der drei 0 am Ende.
(Siehe Colour-Genie-ROM-Listing)

Basicprogramme enthalten keine Prüfsumme, aber durch das Pointersystem (jede Zeile enthält einen Pointer auf die nächste Zeile) kann die geladene Aufzeichnung im gewissen Rahmen geprüft werden, wobei es unerheblich ist, wo das Basicprogramm vorher im Speicher stand.

Beispiel: Demoprogramm

Basicfile: D

Bei Fehlern im Pointersystem wird für jeden falschen Pointer die Meldung Fehler ausgegeben:

Basicfile: D
!!! Fehler !!!
!!! Fehler !!!

Bei beiden Filetypen kann eine weitere Fehlermeldung erscheinen:

File zu kurz.

Diese erscheint zum Beispiel, wenn ein Systemprogramm mitten im Block abbricht oder der Entry fehlt und bei einem Basicprogramm, wenn die drei 0 am Ende fehlen.

Bei Datenaufzeichnungen kann entweder nichts erscheinen oder es kann zufällig als einer der Filetypen erkannt werden. Dann erscheinen höchstwahrscheinlich zahlreiche

Fehlermeldungen.

Fehlermeldungen beeinträchtigen den Kopiervorgang nicht. Sie dienen nur zur Kontrolle.

Eine geladene Aufzeichnung steht bis zum nächsten Laden für Vergleiche und zum Aufzeichnen zur Verfügung

Die Analyse des Files wird bei eingeschaltetem Drucker (<D> im Hauptmenue) paralell zur Bildschirmausgabe auf den Drucker ausgegeben.

Vergleichen

Die in den Speicher geladenen Aufzeichnung kann mit einer weiteren verglichen werden (zum Beispiel eine angefertigte Kopie). Abbruch wie beim Laden und automatisch beim 1. Ladefehler. Beim Vergleichen wird der eingelesene Text nicht auf dem Bildschirm angezeigt. Die Anzeige "FnameBBBBBBBBBE" erscheint, wenn das zu vergleichende File mit "L" geladen wurde, und nicht, wenn es mit "F" geladen wurde.

Schreiben

Kassettenrecorder zur Aufnahme bereit machen und dies bestätigen. Dann erfolgt die Abfrage:

Kopienanzahl ?

Sie können mit <ENTER> oder einer Ziffer antworten. Es wird dann automatisch eine entsprechende Anzahl von Kopien (bei <ENTER> = 1) hintereinander erzeugt. Zwischen 2 Kopien wird automatisch eine Pause von ca. 5 Sekunden eingeschoben. Abbruch des Schreibens ist jederzeit außer während der Pause durch Drücken der <BREAK>-Taste möglich.

Fehlertape laden

Die komplexen Analysevorgänge beim Laden machen das Laden mit COPBIT etwas kritischer als das normale Einladen mit CLOAD oder SYSTEM. Deshalb kann mit <F> eine weitere Laderoutine aufgerufen werden, die während des Laden nur den Text durchlaufen läßt, aber nicht "Fname BBBB" ausdruckt. Dadurch werden viele langwierige Operationen eingespart und das Laden ist erheblich unkritischer. Bei Ladeproblemen helfen im übrigen oft auch folgende Tips:

- 1) Höhen auf Maximum
- 2) Tonkopfjustage: Während der Wiedergabe auf maximale Höhenwiedergabe (deutlich zu hören) abstimmen. Nach erfolgreichem Laden wieder auf Ihre Standardeinstellung mit einer Ihrer alten Kassetten einstellen und dann die Kopien anfertigen. Die COPBIT-Kassette ist übrigens auf einem mit einer Azimut-Normkassette geeichten Kassettenrekorder aufgezeichnet.
- 3) Lautstärke: Bei manchen Aufzeichnungen hilft starkes Übersteuern, bei anderen aber eher eine etwas niedrigere Wahl der Lautstärke (2).

Automatik

Die Automatik (Abbruch des Laden, wenn eine Pause auftritt) kann abgeschaltet werden, wenn eine Aufzeichnung mit Pausen geladen werden soll. Beim Starten des Programmes ist die Automatik ein. Durch Eingabe einer <A> im Hauptmenue kann sie ab- und wieder eingeschaltet werden. Der Zustand des Automatikflags kann im Hauptmenue jederzeit im linken Kasten abgelesen werden.

Drucker

Die Analyse eines geladenen Files kann wahlweise auf einen Drucker zusätzlich ausgedruckt werden. Wenn Sie dies wünschen, müssen Sie im Hauptmenue das <D> antippen. Mit der gleichen Taste können Sie den Drucker wieder abschalten. Der jeweilige Zustand kann im rechten Kasten abgelesen werden. Bitte achten Sie darauf, daß bei eingeschaltetem Drucker, auch wirklich ein Drucker angeschlossen und ON LINE ist. Sonst bleibt das Programm nach dem Laden stehen!

Geschützte Formate

Zahlreiche Programme für das Colour-Genie sind in einem geschützten Format aufgezeichnet, und lassen sich nicht mit einem normalen Kopierprogramm kopieren. Durch das völlig neue Konzept von COPBIT können praktisch alle für das Colour-Genie angebotenen Programme mit hohem Komfort kopiert werden.

Preloader

Viele geschützte Formate arbeiten mit Preloader, d. h. die Aufzeichnung beginnt mit einem kurzen Teil im Systemformat, der einen Autostart (s.o. und ROM-Listing) enthält. Danach kommt ein belangloser Entry und dahinter noch zahlreiche Bytes in einem anderen Format (vom Programmator erfunden), welche von COPBIT als "zus. Bytes" ausgegeben werden. Beim Laden mit dem System-Befehl wird der erste Teil (der Preloader) in den Speicher geladen und nach dem belanglosen Entry wird an die Autostartadresse gesprungen, wo nun der Preloader startet und mit Hilfe der ROM-Routinen den Rest des Programmes im speziellen Format lädt. Dieser Trick läßt die meisten Kopierprogramme scheitern, da Sie den Ladevorgang beim Entry abbrechen. COPBIT aber lädt solange von Kassette bis eine Pause kommt und legt die geladenen Bytes einschließlich aller Steuerzeichen hintereinander unverändert im Speicher ab. Erst nach beendeten Laden wird dort analysiert. Beim Schreiben wird alles wieder genauso auf Kassette gespeichert wie es vorher eingeladen wurde. Preloaderaufzeichnungen lassen sich ohne weiteres mit COPBIT kopieren.

Baudrateumschaltung

Ein anderer weitverbreiteter Kopierschutztrick ist die Veränderung der Baudrate. Beim Colour-Genie können durch Belegen von bestimmten Speicherzellen mit bestimmten Werten die Aufzeichnungs- (4310H + 4311H) und Ladebaudrate (4312H) in weiten Rahmen verändert werden. COPBIT erkennt beim Laden Veränderungen der Baudrate (= Veränderungen der Speicherzelle 4312H) und führt diese entsprechend aus um mit der richtigen Baudrate zu laden. Diese falschen Werte werden aber sofort durch den Standardwert ersetzt im Speicher abgelegt (bei der Analyse können Sie das bei Blöcken folgender Art erkennen: Block : 4312H-4312H : 69 H). Die Baudrateumschaltung können Sie aber auch schon während des Ladevorganges an der Geschwindigkeitsänderung des durchlaufenden Textes erkennen. Die Kopie wird ohne Baudrateumschaltung erstellt, und da alle entsprechenden Blöcke durch die Standardwerte unschädlich gemacht worden sind, ist diese Aufzeichnung trotzdem einwandfrei lauffähig und geringfügig kürzer. Bänder mit Baudrateumschaltung können nicht mit <F> geladen werden, da dann die Analyse, die die Baudrateumschaltung erkennt, entfällt.

```

00100      ; copbit
00110      ;
00120      ; Kopiert Colour-Genie Kassetten Bitweise
00130      ;
00140      ; (c) Luidger Röckrath
00150      ; Noppiusstr. 19;
00160      ; 5100 Aachen
00170      ; (0241)34962
00180      ;
00190      ;
00200      ;
000D      00210 CR EQU ODH
00220      ;
00230      ;
4800      00240 ORG 4800H
00250      ;
00260      ;ASEG
4800 F3    00270 START DI
4801 31694C 00280 LD SP,BUFFER
4804 3E69    00290 LD A,105
4806 321243 00300 LD (4312H),A
4809 3E03    00310 LD A,3
480B CDB03F 00320 CALL 3FB0H
480E 21D9F4 00330 LD HL,TEXT1
4811 CD13F4 00340 CALL PRINT
4814 CD2B00 00350 ST1 CALL INCH
4817 FE61    00360 CP 'a'
4819 3B02    00370 JR C,ST2
481B D620    00380 SUB 'a'-'A'
481D FE4C    00390 ST2 CP 'L'
481F 2B1C    00400 JR Z,LADEN
4821 FE53    00410 CP 'S'
4823 CAC74A 00420 JP Z,SCHREI
4826 FE56    00430 CP 'V'
4828 CA374B 00440 JP Z,VERGLE
482B FE41    00450 CP 'A'
482D CAB44A 00460 JP Z,AUTOM
4830 FE44    00470 CP 'D'
4832 CA00F4 00480 JP Z,DREA
4835 FE46    00490 CP 'F'
4837 CA3C4B 00500 JP Z,FTAPEL
483A 1BDB    00510 JR ST1
00520      ;
00530      ; Fehlerhaftes Tape laden
00540      ;
483C 3E      00550 FTAPEL DEFB 3EH ; = LD A,...
483D AF      00560 LADEN XOR A
483E 32FBF7 00570 LD (FTF),A
4841 21B4F6 00580 LD HL,TEXT2
4844 CD13F4 00590 CALL PRINT
4847 210000 00600 LD HL,0
484A 22EFF7 00610 LD (LSTREC),HL
484D 22EDF7 00620 LD (LSTRAC),HL
4850 CDBF4B 00630 CALL WAIT
4853 21E0F6 00640 LD HL,TEXT6
4856 CD13F4 00650 CALL PRINT
4859 117F46 00660 LD DE,467FH
485C 211CCB 00670 LD HL,0CB1CH
485F 226C4B 00680 LD (READY),HL
4862 FD2A2040 00690 LAD LD IY,(4020H)

```

```

4866 3E55      00700      LD      A,'U'
4868 328648    00710      LD      (L1110+1),A
486B 21684C    00720      LD      HL,BUFFER-1
486E CDF54B    00730      CALL    READS
4871 DA414A    00740      JP      C,L110
4874 3AFBF7    00750      LD      A,(FTF)
4877 A7        00760      AND     A
4878 207D      00770      JR      NZ,L1116
487A CD6C4B    00780 L1     CALL    READY
487D FE3C      00790      CP      3CH
487F 280A      00800      JR      Z,L1111
4881 FE78      00810      CP      78H
4883 2861      00820      JR      Z,L1115
4885 FE55      00830 L1110   CP      55H
4887 2844      00840      JR      Z,L1114
4889 18EF      00850      JR      L1
488B FD360042 00860 L1111   LD      (IY+0),'B'
488F FD23      00870      INC     IY
4891 AF        00880      XOR     A
4892 32FAF7    00890      LD      (CF),A
4895 CD6C4B    00900      CALL    READY
4898 47        00910      LD      B,A
4899 CD6C4B    00920      CALL    READY
489C D9        00930      EXX
489D 6F        00940      LD      L,A
489E D9        00950      EXX
489F CD6C4B    00960      CALL    READY
48A2 D9        00970      EXX
48A3 67        00980      LD      H,A
48A4 D9        00990      EXX
48A5 CD6C4B    01000 L1112   CALL    READY
48A8 D9        01010      EXX
48A9 7C        01020      LD      A,H
48AA FE43      01030      CP      43H
48AC 7D        01040      LD      A,L
48AD 23        01050      INC     HL
48AE D9        01060      EXX
48AF 2010      01070      JR      NZ,L1113
48B1 FE12      01080      CP      12H
48B3 200C      01090      JR      NZ,L1113
48B5 7E        01100      LD      A,(HL)
48B6 321243    01110      LD      (4312H),A
48B9 3669      01120      LD      (HL),105
48BB ED44      01130      NEG
48BD 86        01140      ADD     A,(HL)
48BE 32FAF7    01150      LD      (CF),A
48C1 10E2      01160 L1113   DJNZ    L1112
48C3 CD6C4B    01170      CALL    READY
48C6 3AFBF7    01180      LD      A,(CF)
48C9 86        01190      ADD     A,(HL)
48CA 77        01200      LD      (HL),A
48CB 18AD      01210      JR      L1
48CD FD360046 01220 L1114   LD      (IY+0),'F'
48D1 FD23      01230      INC     IY
48D3 0606      01240      LD      B,6
48D5 CD6C4B    01250 L11114  CALL    READY
48D8 FD7700    01260      LD      (IY+0),A
48DB FD23      01270      INC     IY
48DD 10F6      01280      DJNZ    L11114
48DF 3E3C      01290      LD      A,3CH

```

```

48E1 328648 01300 LD (L1110+1),A
48E4 1894 01310 JR L1
48E6 FD360045 01320 L1115 LD (IY+0),'E'
48EA FD23 01330 INC IY
48EC CD6C4B 01340 CALL READY
48EF CD6C4B 01350 CALL READY
48F2 3E69 01360 LD A,105
48F4 321243 01370 LD (4312H),A
48F7 CD6C4B 01380 L1116 CALL READY
48FA 18FB 01390 JR L1116
48FC 3E00 01400 L22 LD A,0
48FE 8F 01410 ADC A,A
48FF 32F0F7 01420 LD (AFF),A
4902 FD222040 01430 LD (4020H),IY
4906 3AF8F7 01440 LD A,(DF)
4909 32F9F7 01450 LD (DFLAG),A
490C 11694C 01460 LD DE,BUFFER
490F A7 01470 AND A
4910 ED52 01480 SBC HL,DE
4912 E5 01490 PUSH HL
4913 21A9F6 01500 LD HL,TEXT3
4916 CD13F4 01510 CALL PRINT
4917 E1 01520 POP HL
491A CD67F4 01530 CALL PRHEX
491D CDAC4A 01540 CALL CRCR
4920 22EBF7 01550 LD (FLEN),HL
4923 118046 01560 LD DE,46B0H
4926 0680 01570 LD B,128
4928 3E20 01580 LD A,' '
492A 12 01590 L113 LD (DE),A
492B 13 01600 INC DE
492C 10FC 01610 DJNZ L113
492E 11694C 01620 LD DE,BUFFER
4931 2AE8F7 01630 LD HL,(FLEN)
4934 19 01640 ADD HL,DE
4935 2B 01650 DEC HL
4936 3AF0F7 01660 LD A,(AFF)
4939 A7 01670 AND A
493A 2801 01680 JR Z,L114
493C 2B 01690 DEC HL
493D CD984A 01700 L114 CALL NULL
4940 CA4B4A 01710 JP Z,L100
4943 1A 01720 LD A,(DE)
4944 FE55 01730 CP 'U'
4946 2806 01740 JR Z,L115
4948 CD984A 01750 CALL NULL
494B CA4B4A 01760 JP Z,L100
494E 21FFFF 01770 L115 LD HL,OFFFHH
4951 1A 01780 L112 LD A,(DE)
4952 CDBFF4 01790 CALL ENDE
4955 23 01800 INC HL
4956 13 01810 INC DE
4957 FE55 01820 CP 'U'
4959 20F6 01830 JR NZ,L112
495B 7C 01840 LD A,H
495C B5 01850 OR L
495D E5 01860 PUSH HL
495E F5 01870 PUSH AF
495F 216AF7 01880 LD HL,TEXT18
4962 C413F4 01890 CALL NZ,PRINT

```

| | | | |
|---------------|------------|------|-------------|
| 4965 F1 | 01900 | POP | AF |
| 4966 E1 | 01910 | POP | HL |
| 4967 C467F4 | 01920 | CALL | NZ,FRHEX |
| 496A 214EF7 | 01930 | LD | HL,TEXT10 |
| 496D CD13F4 | 01940 | CALL | PRINT |
| 4970 0606 | 01950 | LD | B,6 |
| 4972 1A | 01960 L11 | LD | A,(DE) |
| 4973 CD3BF4 | 01970 | CALL | OUTCH |
| 4976 13 | 01980 | INC | DE |
| 4977 10F9 | 01990 | DJNZ | L11 |
| 4979 21FFFF | 02000 L12 | LD | HL,OFFFHH |
| 497C 1A | 02010 L121 | LD | A,(DE) |
| 497D CDBFF4 | 02020 | CALL | ENDE |
| 4980 13 | 02030 | INC | DE |
| 4981 23 | 02040 | INC | HL |
| 4982 FE78 | 02050 | CP | 78H |
| 4984 2804 | 02060 | JR | Z,L122 |
| 4986 FE3C | 02070 | CP | 3CH |
| 4988 20F2 | 02080 | JR | NZ,L121 |
| 498A F5 | 02090 L122 | PUSH | AF |
| 498B 7C | 02100 | LD | A,H |
| 498C B5 | 02110 | OR | L |
| 498D 280B | 02120 | JR | Z,L123 |
| 498F E5 | 02130 | PUSH | HL |
| 4990 216AF7 | 02140 | LD | HL,TEXT18 |
| 4993 CD13F4 | 02150 | CALL | PRINT |
| 4996 E1 | 02160 | POP | HL |
| 4997 CD67F4 | 02170 | CALL | FRHEX |
| 499A F1 | 02180 L123 | POP | AF |
| 499B FE78 | 02190 | CP | 78H |
| 499D 2873 | 02200 | JR | Z,L15 |
| 499F 1A | 02210 | LD | A,(DE) |
| 49A0 4F | 02220 | LD | C,A |
| 49A1 13 | 02230 | INC | DE |
| 49A2 1A | 02240 | LD | A,(DE) |
| 49A3 6F | 02250 | LD | L,A |
| 49A4 13 | 02260 | INC | DE |
| 49A5 1A | 02270 | LD | A,(DE) |
| 49A6 67 | 02280 | LD | H,A |
| 49A7 13 | 02290 | INC | DE |
| 49A8 85 | 02300 | ADD | A,L |
| 49A9 47 | 02310 | LD | B,A |
| 49AA C5 | 02320 | PUSH | BC |
| 49AB ED53F1F7 | 02330 | LD | (LSTAD),DE |
| 49AF 1A | 02340 L14 | LD | A,(DE) |
| 49B0 CDBFF4 | 02350 | CALL | ENDE |
| 49B3 13 | 02360 | INC | DE |
| 49B4 80 | 02370 | ADD | A,B |
| 49B5 47 | 02380 | LD | B,A |
| 49B6 0D | 02390 | DEC | C |
| 49B7 20F6 | 02400 | JR | NZ,L14 |
| 49B9 1A | 02410 | LD | A,(DE) |
| 49BA 13 | 02420 | INC | DE |
| 49BB 90 | 02430 | SUB | B |
| 49BC 280C | 02440 | JR | Z,L18 |
| 49BE E5 | 02450 | PUSH | HL |
| 49BF CD6FF4 | 02460 | CALL | PRBL |
| 49C2 E1 | 02470 | POP | HL |
| 49C3 010000 | 02480 | LD | BC,0 |
| 49C6 ED43EFF7 | 02490 | LD | (LSTREC),BC |

```

49CA 32F5F7 02500 L18 LD (CSF),A
49CD C1 02510 POP BC
49CE E5 02520 PUSH HL
49CF D5 02530 PUSH DE
49D0 ED5BEFF7 02540 LD DE, (LSTREC)
49D4 7B 02550 LD A,E
49D5 B2 02560 OR D
49D6 2003 02570 JR NZ,L16
49D8 3C 02580 INC A
49D9 180B 02590 JR L17
49DB 37 02600 L16 SCF
49DC ED52 02610 SBC HL,DE
49DE F5 02620 PUSH AF
49DF C46FF4 02630 CALL NZ,PRBL
49E2 F1 02640 POP AF
49E3 D1 02650 L17 POP DE
49E4 E1 02660 POP HL
49E5 280B 02670 JR Z,L13
49E7 22EDF7 02680 LD (LSTRAC),HL
49EA E5 02690 PUSH HL
49EB 2AF1F7 02700 LD HL,(LSTAD)
49EE 22F3F7 02710 LD (LSTAD1),HL
49F1 E1 02720 POP HL
49F2 0600 02730 L13 LD B,0
49F4 0D 02740 DEC C
49F5 09 02750 ADD HL,BC
49F6 22EFF7 02760 LD (LSTREC),HL
49F9 3AF5F7 02770 LD A,(CSF)
49FC A7 02780 AND A
49FD CA7949 02790 JP Z,L12
4A00 CD6FF4 02800 CALL PRBL
4A03 210000 02810 LD HL,0
4A06 22EFF7 02820 LD (LSTREC),HL
4A09 2186F7 02830 LD HL,TEXT13
4A0C CD13F4 02840 CALL PRINT
4A0F C37949 02850 JP L12
4A12 CD6FF4 02860 ;
4A15 2178F7 02870 L15 CALL PRBL
4A18 CD13F4 02880 LD HL,TEXT12
4A1B 1A 02890 CALL PRINT
4A1C 6F 02900 LD A,(DE)
4A1D 13 02910 LD L,A
4A1E 1A 02920 INC DE
4A1F 67 02930 LD A,(DE)
4A20 CD67F4 02940 LD H,A
4A23 13 02950 CALL PRHEX
4A24 21694C 02960 INC DE
4A27 ED4BEBF7 02970 LD HL,BUFFER
4A2B 09 02980 LD BC,(FLEN)
4A2C A7 02990 ADD HL,BC
4A2D ED52 03000 AND A
4A2F 7D 03010 SBC HL,DE
4A30 B4 03020 LD A,L
4A31 280B 03030 OR H
4A33 E5 03040 JR Z,L111
4A34 216AF7 03050 PUSH HL
4A37 CD13F4 03060 LD HL,TEXT18
4A3A E1 03070 CALL PRINT
4A3B CD67F4 03080 POP HL
4A3C 03090 CALL PRHEX

```

```

4A3E CDAC4A 03100 L111 CALL CRCR
4A41 AF 03110 L110 XOR A
4A42 32F9F7 03120 LD (DFLAG),A
4A45 CDBF4B 03130 WSTART CALL WAIT
4A48 C30048 03140 JP START
03150 ;
4A4B 2195F7 03160 L100 LD HL,TEXT14
4A4E CD13F4 03170 CALL PRINT
4A51 1A 03180 LD A,(DE)
4A52 CD3BF4 03190 CALL OUTCH
4A55 13 03200 INC DE
4A56 CDAC4A 03210 CALL CRCR
4A59 EB 03220 EX DE,HL
4A5A 5E 03230 L105 LD E,(HL)
4A5B 23 03240 INC HL
4A5C 56 03250 LD D,(HL)
4A5D 23 03260 INC HL
4A5E 23 03270 INC HL
4A5F 23 03280 INC HL
4A60 7E 03290 L101 LD A,(HL)
4A61 CDB9F4 03300 CALL ENDE1
4A64 23 03310 INC HL
4A65 A7 03320 AND A
4A66 20F8 03330 JR NZ,L101
4A68 CDB9F4 03340 L102 CALL ENDE1
4A6B D5 03350 PUSH DE
4A6C 5E 03360 LD E,(HL)
4A6D 23 03370 INC HL
4A6E 56 03380 LD D,(HL)
4A6F 7A 03390 LD A,D
4A70 B3 03400 OR E
4A71 E3 03410 EX (SP),HL
4A72 EB 03420 EX DE,HL
4A73 C1 03430 POP BC
4A74 CA414A 03440 JP Z,L110
4A77 E5 03450 PUSH HL
4A78 ED52 03460 SBC HL,DE
4A7A 09 03470 ADD HL,BC
4A7B D1 03480 POP DE
4A7C 2B 03490 DEC HL
4A7D 2B 03500 DEC HL
4A7E 7E 03510 LD A,(HL)
4A7F 23 03520 INC HL
4A80 A7 03530 AND A
4A81 2BE5 03540 JR Z,L102
4A83 E5 03550 PUSH HL
4A84 2186F7 03560 LD HL,TEXT13
4A87 CD13F4 03570 CALL PRINT
4A8A E1 03580 POP HL
4A8B CDAF4A 03590 CALL CR1
4A8E 7E 03600 L103 LD A,(HL)
4A8F CDB9F4 03610 CALL ENDE1
4A92 23 03620 INC HL
4A93 A7 03630 AND A
4A94 20F8 03640 JR NZ,L103
4A96 18C2 03650 JR L105
03660 ;
4A98 7E 03670 NULL LD A,(HL)
4A99 2B 03680 DEC HL
4A9A B6 03690 OR (HL)

```

```

4A9B 2B      03700      DEC      HL
4A9C B6      03710      OR       (HL)
4A9D 23      03720      INC      HL
4A9E C0      03730      RET      NZ
4A9F 2B      03740      DEC      HL
4AA0 2B      03750      DEC      HL
4AA1 7E      03760      LD       A, (HL)
4AA2 FE78    03770      CP       78H
4AA4 23      03780      INC      HL
4AA5 23      03790      INC      HL
4AA6 2B02    03800      JR       Z, N1
4AA8 AF      03810      XOR      A
4AA9 C9      03820      RET
4AAA A7      03830 N1    AND      A
4AAB C9      03840      RET
              03850      ;
4AAC CDAF4A  03860 CRCR  CALL    CR1
4AAF 3E0D    03870 CR1  LD       A, CR
4AB1 C33BF4  03880      JP       OUTCH
              03890      ;
              03900      ; Automatik (Ein/Aus)
              03910      ;
4AB4 3AF6F7  03920 AUTOM LD      A, (BF)
4AB7 EEFF    03930      XOR      255
4AB9 32F6F7  03940      LD      (BF), A
4ABC 3A79F5  03950      LD      A, (BA)
4ABF EE61    03960      XOR      'a' ; ' ' XOR 'A'
4AC1 3279F5  03970      LD      (BA), A
4AC4 C30048  03980      JP      START
              03990      ;
              04000      ; Schreiben
              04010      ;
4AC7 21BDF6  04020 SCHREI LD      HL, TEXT4
4ACA CD13F4  04030      CALL    PRINT
4ACD CD8F4B  04040      CALL    WAIT
4AD0 21BAF7  04050      LD      HL, TEXT16
4AD3 CD13F4  04060      CALL    PRINT
4AD6 CD2B00  04070 S0    CALL    INCH
4AD9 FE0D    04080      CP      CR
4ADB 0601    04090      LD      B, 1
4ADD 2B09    04100      JR      Z, S01
4ADF D630    04110      SUB     '0'
4AE1 3BF3    04120      JR      C, S0
4AE3 FE0A    04130      CP      10
4AE5 30EF    04140      JR      NC, S0
4AE7 47      04150      LD      B, A
4AEB 7B      04160 S01   LD      A, B
4AE9 C630    04170      ADD     A, '0'
4AEB CD3BF4  04180      CALL    OUTCH
4AEE 4B      04190      LD      C, B
4AEF 21F2F6  04200      LD      HL, TEXT7
4AF2 CD13F4  04210      CALL    PRINT
4AF5 C5      04220 S11   PUSH    BC
4AF6 79      04230      LD      A, C
4AF7 90      04240      SUB     B
4AF8 C631    04250      ADD     A, '1'
4AFA 32CBF7  04260      LD      (TEXT17), A
4AFD F5      04270      PUSH    AF
4AFE 21CBF7  04280      LD      HL, TEXT17
4B01 CD13F4  04290      CALL    PRINT

```



```

4B04 F1      04300      POP      AF
4B05 FE31    04310      CP       '1'
4B07 280B    04320      JR       Z,S13
4B09 1605    04330      LD       D,5
4B0B 010000  04340 S12    LD       BC,0
4B0E CD6000  04350      CALL     60H
4B11 15      04360      DEC      D
4B12 20F7    04370      JR       NZ,S12
4B14 CD9D4B  04380 S13    CALL     WRITES
4B17 201B    04390      JR       NZ,S2
4B19 21694C  04400      LD       HL,BUFFER
4B1C ED4BEBF7 04410      LD       BC,(FLEN)
4B20 7E      04420 S1    LD       A,(HL)
4B21 CD1F02  04430      CALL     WRITEB
4B24 3A40F8  04440      LD       A,(0F840H)
4B27 CB57    04450      BIT      2,A
4B29 2009    04460      JR       NZ,S2
4B2B 23      04470      INC      HL
4B2C 0B      04480      DEC      BC
4B2D 7B      04490      LD       A,B
4B2E B1      04500      OR       C
4B2F 20EF    04510      JR       NZ,S1
4B31 C1      04520      POP      BC
4B32 10C1    04530      DJNZ     S11
4B34 C3454A  04540 S2    JP       WSTART
                04550      ;
                04560      ; Vergleichen
                04570      ;
4B37 2194F6  04580 VERGLE LD      HL,TEXT2
4B3A CD13F4  04590      CALL     PRINT
4B3D CDBF4B  04600      CALL     WAIT
4B40 210FF7  04610      LD       HL,TEXTB
4B43 CD13F4  04620      CALL     PRINT
4B46 260F    04630      LD       H,READY-READY-2
4B48 2E18    04640      LD       L,18H
4B4A 226C4B  04650      LD       (READY),HL
4B4D ED5BEBF7 04660      LD       DE,(FLEN)
4B51 C3624B  04670      JP       LAD
                04680      ;
4B54 21A2F7  04690 V1    LD      HL,TEXT15
4B57 FD222040 04700 V4    LD      (4020H),IV
4B5B E5      04710      PUSH     HL
4B5C CDAC4A  04720      CALL     CROR
4B5F E1      04730      POP      HL
4B60 CD13F4  04740      CALL     PRINT
4B63 D1      04750 V2    POP      DE
4B64 C3454A  04760      JP       WSTART
                04770      ;
                04780      ; Fehler
                04790      ;
4B67 21A7F7  04800 V3    LD      HL,TEXT5
4B6A 18EB    04810      JR       V4
                04820      ;
                04830      ;
                04840      ; Unterprogramme für Lesen
                04850      ;
4B6C 1C      04860 READY INC      E
4B6D CBF8    04870      SET      7,E
4B6F CDB24B  04880      CALL     READB
4B72 2804    04890      JR       Z,REND

```

```

4B74 23      04900      INC      HL
4B75 77      04910      LD       (HL),A
4B76 12      04920      LD       (DE),A
4B77 D0      04930      RET
4B78 C1      04940 REND  POP      BC
4B79 23      04950      INC      HL
4B7A C3FC48  04960      JP       L22
                04970 ;
4B7D 1B      04980 READV DEC      DE
4B7E 7A      04990      LD       A,D
4B7F B3      05000      OR       E
4B80 28D2    05010      JR       Z,V1
4B82 CDB24B  05020      CALL     READB
4B85 23      05030      INC      HL
4B86 28DB    05040      JR       Z,V2
4B88 28D9    05050      JR       C,V2
4B8A BE      05060      CP       (HL)
4B8B C8      05070      RET      Z
4B8C C3674B  05080      JP       V3
                05090 ;
                05100 ; Unterprogramme
                05110 ;
4B8F 212DF7  05120 WAIT  LD       HL,TEXT0
4B92 CD13F4  05130      CALL     PRINT
4B95 CD2B00  05140 W1    CALL     INCH
4B98 FE0D    05150      CP       CR
4B9A 20F9    05160      JR       NZ,W1
4B9C C9      05170      RET
                05180 ;
4B9D 06FF    05190 WRITES LD      B,255
4B9F 3EAA    05200 WR1   LD      A,0AAH
4BA1 CD1F02  05210      CALL     WRITEB
4BA4 3A40F8  05220      LD      A,(0F840H)
4BA7 CB57    05230      BIT      2,A
4BA9 C0      05240      RET      NZ
4BAA 10F3    05250      DJNZ     WR1
4BAC AF      05260      XOR      A
4BAD 3E66    05270      LD      A,66H
4BAF C31F02  05280      JP       WRITEB
                05290 ;
021F         05300 WRITEB EQU     021FH
                05310 ;
4BB2 D9      05320 READB EXX
4BB3 0608    05330      LD      B,B
4BB5 1600    05340      LD      D,0
4BB7 CDCF4B  05350 RB1   CALL     READBI
4BBA 3B06    05360      JR       C,RB2
4BBC 10F9    05370      DJNZ     RB1
4BBE 04      05380      INC      B
4BBF 7A      05390 RB3   LD      A,D
4BC0 D9      05400      EXX
4BC1 C9      05410      RET
                05420 ;
4BC2 78      05430 RB2   LD      A,B
4BC3 FE08    05440      CP      B
4BC5 28F8    05450      JR       Z,RB3
4BC7 CB22    05460 RB4   SLA      D
4BC9 10FC    05470      DJNZ     RB4
4BCB 37      05480      SCF
4BCC 04      05490      INC      B

```

```

4BCD 18F0      05500      JR      RB3
                05510      ;
4BCF C5        05520 READBI PUSH    BC
4BD0 DBFF      05530      IN      A, (OFFH)
4BD2 E601      05540      AND     01
4BD4 5F        05550      LD      E, A
4BD5 0E00      05560      LD      C, 0
4BD7 0D        05570 R1     DEC     C
4BD8 280B      05580      JR      Z, RBI1
4BDA DBFF      05590      IN      A, (OFFH)
4BDC E601      05600      AND     01
4BDE AB        05610      XOR     E
4BDF 1F        05620      RRA
4BE0 30F5      05630      JR      NC, R1
4BE2 C30B02    05640      JP      020BH
4BE5 3AF6F7    05650 RBI1   LD      A, (BF)
4BE8 A7        05660      AND     A
4BE9 2807      05670      JR      Z, RBI2
4BEB 3A40F8    05680      LD      A, (0F840H)
4BEE CB57      05690      BIT     2, A
4BF0 28E5      05700      JR      Z, R1
4BF2 C1        05710 RBI2   POP     BC
4BF3 37        05720      SCF
4BF4 C9        05730      RET
                05740      ;
4BF5 3AF6F7    05750 READS  LD      A, (BF)
4BF8 32F7F7    05760      LD      (BFS), A
4BFB 3E01      05770      LD      A, 1
4BFD 32F6F7    05780      LD      (BF), A
4C00 E5        05790      PUSH    HL
4C01 D5        05800      PUSH    DE
4C02 C5        05810      PUSH    BC
4C03 3E05      05820      LD      A, 5
4C05 3226F0    05830      LD      (0F026H), A
4C08 3227F0    05840      LD      (0F027H), A
4C0B 01AA80    05850 RS1    LD      BC, 0B0AAH
4C0E CDCF4B    05860 RS2    CALL    READBI
4C11 381A      05870      JR      C, RS3
4C13 B9        05880      CP      C
4C14 20F5      05890      JR      NZ, RS1
4C16 3EFF      05900      LD      A, 255
4C18 A9        05910      XOR     C
4C19 4F        05920      LD      C, A
4C1A 10F2      05930      DJNZ    RS2
4C1C CDCF4B    05940 RS4    CALL    READBI
4C1F 380C      05950      JR      C, RS3
4C21 FE66      05960      CP      66H
4C23 20F7      05970      JR      NZ, RS4
4C25 3E2A      05980      LD      A, '*'
4C27 322644    05990      LD      (4426H), A
4C2A 322744    06000      LD      (4427H), A
4C2D C1        06010 RS3    POP     BC
4C2E D1        06020      POP     DE
4C2F E1        06030      POP     HL
4C30 3AF7F7    06040      LD      A, (BFS)
4C33 32F6F7    06050      LD      (BF), A
4C36 C9        06060      RET
                06070      ;
0032          06080      DEFS    50
4C69          06090 BUFFER EQU    $

```

```

                                06100      ;
F400      06110      ORG      0F400H
                                06120      ;
                                06130      ; Drucker (Ein/Aus)
                                06140      ;
F400 3AF8F7 06150 DREA      LD      A, (DF)
F403 EEFF   06160      XOR      255
F405 32F8F7 06170      LD      (DF), A
F408 3A7FF5 06180      LD      A, (DA)
F40B EE64   06190      XOR      'd' ; ' ' XOR 'D'
F40D 327FF5 06200      LD      (DA), A
F410 C30048 06210      JP      START
                                06220      ;
F413 7E     06230 PRINT    LD      A, (HL)
F414 A7     06240      AND      A
F415 23     06250      INC      HL
F416 2B09   06260      JR      Z, PR2
F418 FEFF   06270      CP      255
F41A 2812   06280      JR      Z, PR3
F41C CD3BF4 06290      CALL    OUTCH
F41F 1BF2   06300      JR      PRINT
                                06310      ;
F421 D5     06320 PR2     PUSH    DE
F422 CD2B00 06330      CALL    INCH
F425 FE60   06340      CP      60H
F427 CC2B00 06350 PR1    CALL    Z, INCH
F42A 28FB   06360      JR      Z, PR1
F42C D1     06370      POP     DE
F42D C9     06380      RET
                                06390      ;
F42E C5     06400 PR3    PUSH    BC
F42F 46     06410      LD      B, (HL)
F430 23     06420      INC      HL
F431 7E     06430 PR4    LD      A, (HL)
F432 CD3BF4 06440      CALL    OUTCH
F435 10FA   06450      DJNZ    PR4
F437 23     06460      INC      HL
F438 C1     06470      POP     BC
F439 18D8   06480      JR      PRINT
                                06490      ;
002B      06500 INCH     EQU     002BH
                                06510      ;
F43B D5     06520 OUTCH   PUSH    DE
F43C F5     06530      PUSH    AF
F43D CD3300 06540      CALL    33H
F440 F1     06550      POP     AF
F441 57     06560      LD      D, A
F442 3AF9F7 06570      LD      A, (DFLAG)
F445 A7     06580      AND      A
F446 7A     06590      LD      A, D
F447 C43B00 06600      CALL    NZ, 3BH
F44A D1     06610      POP     DE
F44B C9     06620      RET
                                06630      ;
                                06640      ;
F44C 7C     06650 PRH    LD      A, H
F44D CD51F4 06660      CALL    PRBYTE
F450 7D     06670      LD      A, L
F451 F5     06680 PRBYTE  PUSH    AF
F452 0F     06690      RRCA

```

| | | | |
|-------------|-------------|------|-------------|
| F453 0F | 06700 | RRCA | |
| F454 0F | 06710 | RRCA | |
| F455 0F | 06720 | RRCA | |
| F456 CD5AF4 | 06730 | CALL | PRNIB |
| F459 F1 | 06740 | POP | AF |
| F45A E60F | 06750 PRNIB | AND | 0FH |
| F45C FE0A | 06760 | CP | 10 |
| F45E 3802 | 06770 | JR | C,PN1 |
| F460 C607 | 06780 | ADD | A,'A'-'9'-1 |
| F462 C630 | 06790 PN1 | ADD | A,'0' |
| F464 C33BF4 | 06800 | JP | OUTCH |
| | 06810 | : | |
| F467 CD4CF4 | 06820 PRHEX | CALL | PRH |
| F46A 3E48 | 06830 | LD | A,'H' |
| F46C C33BF4 | 06840 | JP | OUTCH |
| | 06850 | : | |
| F46F 2AEFF7 | 06860 PRBL | LD | HL,(LSTREC) |
| F472 7D | 06870 | LD | A,L |
| F473 B4 | 06880 | OR | H |
| F474 C8 | 06890 | RET | Z |
| F475 D5 | 06900 | PUSH | DE |
| F476 215CF7 | 06910 | LD | HL,TEXT11 |
| F479 CD13F4 | 06920 | CALL | PRINT |
| F47C 2AEDF7 | 06930 | LD | HL,(LSTRAC) |
| F47F E5 | 06940 | PUSH | HL |
| F480 CD67F4 | 06950 | CALL | PRHEX |
| F483 3E2D | 06960 | LD | A,'-' |
| F485 CD3BF4 | 06970 | CALL | OUTCH |
| F488 2AEFF7 | 06980 | LD | HL,(LSTREC) |
| F48B CD67F4 | 06990 | CALL | PRHEX |
| F48E D1 | 07000 | POP | DE |
| F48F A7 | 07010 | AND | A |
| F490 ED52 | 07020 | SBC | HL,DE |
| F492 D1 | 07030 | POP | DE |
| F493 7C | 07040 | LD | A,H |
| F494 A7 | 07050 | AND | A |
| F495 C0 | 07060 | RET | NZ |
| F496 7D | 07070 | LD | A,L |
| F497 FE03 | 07080 | CP | 3 |
| F499 D0 | 07090 | RET | NC |
| F49A C5 | 07100 | PUSH | BC |
| F49B 45 | 07110 | LD | B,L |
| F49C 04 | 07120 | INC | B |
| F49D 21E7F7 | 07130 | LD | HL,TEXT20 |
| F4A0 CD13F4 | 07140 | CALL | PRINT |
| F4A3 2AF3F7 | 07150 | LD | HL,(LSTAD1) |
| F4A6 7E | 07160 PRBL1 | LD | A,(HL) |
| F4A7 CD51F4 | 07170 | CALL | PRBYTE |
| F4AA 3E48 | 07180 | LD | A,'H' |
| F4AC CD3BF4 | 07190 | CALL | OUTCH |
| F4AF 3E20 | 07200 | LD | A,' ' |
| F4B1 CD3BF4 | 07210 | CALL | OUTCH |
| F4B4 23 | 07220 | INC | HL |
| F4B5 10EF | 07230 | DJNZ | PRBL1 |
| F4B7 C1 | 07240 | POP | BC |
| F4B8 C9 | 07250 | RET | |
| | 07260 | : | |
| | 07270 | : | |
| | 07280 | : | |
| F4B9 EB | 07290 ENDE1 | EX | DE,HL |

```

F4BA CDBFF4 07300 CALL ENDE
F4BD EB 07310 EX DE,HL
F4BE C9 07320 RET
07330 ;
F4BF E5 07340 ENDE PUSH HL
F4C0 D5 07350 PUSH DE
F4C1 21694C 07360 LD HL,BUFFER
F4C4 ED5BEBF7 07370 LD DE,(FLEN)
F4C8 19 07380 ADD HL,DE
F4C9 37 07390 SCF
F4CA D1 07400 POP DE
F4CB ED52 07410 SEC HL,DE
F4CD E1 07420 POP HL
F4CE D0 07430 RET NC
F4CF D1 07440 POP DE
F4D0 21D6F7 07450 LD HL,TEXT19
F4D3 CD13F4 07460 CALL PRINT
F4D6 C3414A 07470 JP L110
07480 ;
07490 ;
07500 ; Texte
07510 ;
F4D9 1C 07520 TEXT1 DEFB 1CH
F4DA 1F 07530 DEFB 1FH
F4DB 43 07540 DEFM 'COPBIT V3.0 Kassettenkopierprogramm'
F4FE 0D 07550 DEFB CR
F4FF 0D 07560 DEFB CR
F500 4B 07570 DEFM 'Kopiert beliebige Kassetten bitweise.'
F525 0D 07580 DEFB CR
F526 0D 07590 DEFB CR
F527 28 07600 DEFM '(C) 1984 Luidger Roeckrath, 5100 Aachen'
F54E 0D 07610 DEFB CR
F54F 20 07620 DEFB ' '
F550 FF 07630 DEFB 255
F551 26 07640 DEFB 38
F552 B4 07650 DEFB 180
F553 20 07660 DEFB ' '
F554 DB 07670 DEFB 219
F555 FF 07680 DEFB 255
F556 1A 07690 DEFB 26
F557 20 07700 DEFB ' '
F558 DB 07710 DEFB 219
F559 20 07720 DEFM ' '
F55E 8E 07730 DEFB 142
F55F 20 07740 DEFM ' '
F564 CF 07750 DEFB 207
F565 DB 07760 DEFB 219
F566 20 07770 DEFM ' Funktionen:'
F573 FF 07780 DEFB 255
F574 0D 07790 DEFB 13
F575 20 07800 DEFB ' '
F576 DB 07810 DEFB 219
F577 20 07820 DEFM ' '
F579 41 07830 BA DEFM 'A '
F57C 8E 07840 DEFB 142
F57D 20 07850 DEFM ' '
F57F 20 07860 DA DEFM ' '
F582 CF 07870 DEFB 207
F583 DB 07880 DEFB 219
F584 FF 07890 DEFB 255

```

```

F585 1A      07900      DEFB      26
F586 20      07910      DEFB      ' '
F587 DB      07920      DEFB      219
F588 20      07930      DEFM      ' '
F58D 8E      07940      DEFB      142
F58E 20      07950      DEFM      ' '
F593 CF      07960      DEFB      207
F594 DB      07970      DEFB      219
F595 FF      07980      DEFB      255
F596 26      07990      DEFB      38
F597 B5      08000      DEFB      181
F598 CF      08010      DEFB      207
F599 DB      08020      DEFB      219
F59A FF      08030      DEFB      255
F59B 26      08040      DEFB      38
F59C 20      08050      DEFB      ' '
F59D CF      08060      DEFB      207
F59E DB      08070      DEFB      219
F59F 20      08080      DEFM      ' <L> a d e n '
F5AC FF      08090      DEFB      255
F5AD 19      08100      DEFB      25
F5AE 20      08110      DEFB      ' '
F5AF CF      08120      DEFB      207
F5B0 DB      08130      DEFB      219
F5B1 FF      08140      DEFB      255
F5B2 26      08150      DEFB      38
F5B3 20      08160      DEFB      ' '
F5B4 CF      08170      DEFB      207
F5B5 DB      08180      DEFB      219
F5B6 20      08190      DEFM      ' <V> e r g l e i c h e n '
F5CF FF      08200      DEFB      255
F5D0 0D      08210      DEFB      13
F5D1 20      08220      DEFB      ' '
F5D2 CF      08230      DEFB      207
F5D3 DB      08240      DEFB      219
F5D4 FF      08250      DEFB      255
F5D5 26      08260      DEFB      38
F5D6 20      08270      DEFB      ' '
F5D7 CF      08280      DEFB      207
F5D8 DB      08290      DEFB      219
F5D9 20      08300      DEFM      ' <S> c h r e i b e n '
F5EE FF      08310      DEFB      255
F5EF 11      08320      DEFB      17
F5F0 20      08330      DEFB      ' '
F5F1 CF      08340      DEFB      207
F5F2 DB      08350      DEFB      219
F5F3 FF      08360      DEFB      255
F5F4 26      08370      DEFB      38
F5F5 20      08380      DEFB      ' '
F5F6 CF      08390      DEFB      207
F5F7 DB      08400      DEFB      219
F5F8 20      08410      DEFM      ' <F> e h l e r t a p e   l a d e n   '
F61E CF      08420      DEFB      207
F61F DB      08430      DEFB      219
F620 FF      08440      DEFB      255
F621 26      08450      DEFB      38
F622 20      08460      DEFB      ' '
F623 CF      08470      DEFB      207
F624 DB      08480      DEFB      219
F625 20      08490      DEFM      ' <A> u t o m a t i k   ( E i n / A u s ) '

```

```

F64B CF      08500      DEFB      207
F64C DB      08510      DEFB      219
F64D FF      08520      DEFB      255
F64E 26      08530      DEFB      38
F64F 20      08540      DEFB      ' '
F650 CF      08550      DEFB      207
F651 DB      08560      DEFB      219
F652 20      08570      DEFM      ' <D>r u c k e r   ( E i n / A u s ) '
F678 CF      08580      DEFB      207
F679 DB      08590      DEFB      219
F67A FF      08600      DEFB      255
F67B 26      08610      DEFB      38
F67C 20      08620      DEFB      ' '
F67D CF      08630      DEFB      207
F67E 20      08640      DEFB      ' '
F67F FF      08650      DEFB      255
F680 26      08660      DEFB      38
F681 B5      08670      DEFB      181
F682 20      08680      DEFB      ' '
F683 00      08690      DEFB      0
F684 1C      08700 TEXT2  DEFB      1CH
F685 1F      08710      DEFB      1FH
F686 4B      08720      DEFM      'Kassettenrecorder auf Wiedergabe'
F6A6 0D      08730      DEFB      CR
F6A7 0D      08740      DEFB      CR
F6A8 00      08750      DEFB      0
F6A9 0D      08760 TEXT3  DEFB      CR
F6AA 0D      08770      DEFB      CR
F6AB 4C      08780      DEFM      'Laenge des Files:'
F6BC 00      08790      DEFB      0
F6BD 1C      08800 TEXT4  DEFB      1CH
F6BE 1F      08810      DEFB      1FH
F6BF 4B      08820      DEFM      'Kassettenrecorder auf Aufnahme'
F6DD 0D      08830      DEFB      CR
F6DE 0D      08840      DEFB      CR
F6DF 00      08850      DEFB      0
F6E0 4C      08860 TEXT6  DEFM      'L a d e n . . . '
F6EF 0D      08870 TEXT6A DEFB      CR
F6F0 0D      08880      DEFB      CR
F6F1 00      08890      DEFB      0
F6F2 0D      08900 TEXT7  DEFB      CR
F6F3 0D      08910      DEFB      CR
F6F4 53      08920      DEFM      'S c h r e i b e n . . . '
F70B 0F      08930      DEFB      15
F70C 0D      08940      DEFB      CR
F70D 0D      08950      DEFB      CR
F70E 00      08960      DEFB      0
F70F 56      08970 TEXT8  DEFM      'V e r g l e i c h e n . . . '
F72A 0D      08980      DEFB      CR
F72B 0D      08990      DEFB      CR
F72C 00      09000      DEFB      0
F72D 57      09010 TEXT0  DEFM      'Weiter. Bitte <ENTER> druecken'
F74B 0D      09020      DEFB      CR
F74C 0D      09030      DEFB      CR
F74D 00      09040      DEFB      0
F74E 0D      09050 TEXT10 DEFB      CR
F74F 53      09060      DEFM      'Systemfile: '
F75B 00      09070      DEFB      0
F75C 0D      09080 TEXT11 DEFB      CR
F75D 42      09090      DEFM      'Block      : '

```



```

F769 00      09100      DEFB      0
F76A 0D      09110 TEXT18 DEFB      CR
F76B 7A      09120      DEFM      'zus. Bytes: '
F777 00      09130      DEFB      0
F778 0D      09140 TEXT12 DEFB      CR
F779 45      09150      DEFM      'Entry      : '
F785 00      09160      DEFB      0
F786 21      09170 TEXT13 DEFM      '!!! Fehler !!!'
F794 00      09180      DEFB      0
F795 42      09190 TEXT14 DEFM      'Basicfile : '
F7A1 00      09200      DEFB      0
F7A2 4B      09210 TEXT15 DEFM      'Kein '
F7A7 56      09220 TEXT5  DEFM      'Vergleichsfehler'
F7B7 0D      09230      DEFB      CR
F7B8 0D      09240      DEFB      CR
F7B9 00      09250      DEFB      0
F7BA 4B      09260 TEXT16 DEFM      'Kopienanzahl ? '
F7C9 0E      09270      DEFB      14
F7CA 00      09280      DEFB      0
F7CB 30      09290 TEXT17 DEFM      '0. Kopie'
F7D3 0D      09300      DEFB      CR
F7D4 0D      09310      DEFB      CR
F7D5 00      09320      DEFB      0
F7D6 0D      09330 TEXT19 DEFB      CR
F7D7 46      09340      DEFM      'File zu kurz.'
F7E4 0D      09350      DEFB      CR
F7E5 0D      09360      DEFB      CR
F7E6 00      09370      DEFB      0
F7E7 20      09380 TEXT20 DEFM      ' : '
F7EA 00      09390      DEFB      0
F7EB 0000     09400 FLEN   DEFW      0
F7ED 0000     09410 LSTRAC DEFW      0
F7EF 0000     09420 LSTREC DEFW      0
F7F1 0000     09430 LSTAD  DEFW      0
F7F3 0000     09440 LSTAD1 DEFW      0
F7F5 00      09450 CSF    DEFB      0
F7F6 00      09460 BF     DEFB      0
F7F7 00      09470 BFS    DEFB      0
F7F8 00      09480 DF     DEFB      0
F7F9 00      09490 DFLAG  DEFB      0
F7FA 00      09500 CF     DEFB      0
F7FB 00      09510 FTF    DEFB      0
F7FC 00      09520 AFF    DEFB      0
4800      09530      END      START
00000 TOTAL ERRORS
21954 TEXT AREA BYTES LEFT

```

2.5.1 Kopierprogramm SAVETAPE

=====

SAVETAPE ermöglicht Kopieren fast aller Maschinenprogramme von Kassette auf Diskette. Maschinenprogramme, die nicht mit Speicherbereichen kollidieren, die das System benötigt, können unverändert übernommen werden, andere können verschoben werden und werden mit einem Zusatz versehen, der nach dem Laden das DOS abschaltet und das Programm in den richtigen Speicherbereich verschiebt.

Funktion

Savetape lädt von Kassette wie COPBIT (siehe dort). Im Speicher wird das Programm abgelegt, wie es auf der Kassette mit allen Headern aufgezeichnet war (siehe (2): Teil 1.7). Bei der Aufzeichnung auf Diskette muß das Format in das DOS-Loaderformat (siehe Teil 1, Kapitel 3.5) umgewandelt werden. Dies geschieht in der großen Schleife, die bei S1 beginnt. Blöcke mit der Anfangsadresse 41E2H werden abgefangen, und die Adresse des Sprungbefehles in AE gerettet. Nach dem Entrypointblock wird der Entrypoint durch diese Autostartadresse ersetzt.

Muß das File verschoben werden, so wird das gesamte File so auf Diskette aufgezeichnet, daß es später in den Speicherbereich von der Verschiebeadresse an aufwärts geladen wird. Für jeden Block wird im Anhang die richtige Adresse und die Länge notiert. Nach dem Laden wird zunächst der Anhang (VERSCH) gestartet, der zuerst das DOS abschaltet, indem er eine Initialisierung des Kassettenbasics simuliert. Dann wird ein Block nach dem anderen an seine richtige Adresse verschoben. Autostartblöcke werden beim verschobenen Aufzeichnen nicht wie oben behandelt sondern ganz normal verarbeitet. Beim Verteilen der Blöcke nach dem Laden durch den Anhang wird also ein Autostartblock an seine richtige Speicheradresse übertragen. Damit ein Autostart wirksam wird, ruft der Anhang nach Verschieben des gesamten Programmes die Adresse 41E2H per CALL auf. Das gleiche tut der SYSTEM-Befehl nach Laden einer Aufzeichnung. Steht dort ein RET, wird nach Rückkehr vom Unterprogrammaufruf die richtige Startadresse angesprungen.

Da SAVETAPE sehr umfangreich ist, mußte das Quellprogramm für den COLASM etwas beschnitten werden. Dazu wurden alle Texte stark gekürzt. Die Originaltexte sind vor dem Quellprogrammlisting angegeben. Außerdem müssen alle Kommentar und alle Leerzeichen, die syntaktisch nicht unbedingt notwendig sind gestrichen, werden um mit dem geringen Speicher, den der COLASM übrig läßt auszukommen. Auf der Kassette ist SAVETAPE in dieser Form enthalten.

Technische Anmerkung:

SAVETAPE benutzt verschiedene Speicherbereiche, die im Colour-Genie-System eigentlich für bestimmte Systemaufgaben reserviert sind. Ein Teil von SAVETAPE ist im Speicher für die freiprogrammierbaren Zeichen untergebracht, wodurch manchmal kleine Bildschirmstörungen auftreten. Durch diese intensive Ausnutzung des Speicher war es aber möglich 25736 (=6488H im 32K-System, im 16K-System: 9352 = 2488H) Bytes für das zu kopierende File zur Verfügung zu stellen.

Hauptmenue

Nachdem SAVETAPE geladen ist, meldet sich das Hauptmenue:

***** SAVETAPE V1.0 *****

Kopiert Kassettenprogramme auf Diskette.

(C) 1984 Luidger Roeckrath, 5100 Aachen

```

+-----+-----+-----+
I Funktionen:          I A I      I
+-----+-----+-----+
I                               I
I <L> a d e n           I
I                               I
I <V> e r g l e i c h e n       I
I                               I
I <S> p e i c h e r n a u f D i s k I
I                               I
I <F> e h l e r t a p e l a d e n I
I                               I
I <A> u t o m a t i k (E i n / A u s)I
I                               I
I <D> r u c k e r (E i n / A u s)  I
I                               I
+-----+-----+-----+

```

Durch Antippen der Buchstaben in <> kann die entsprechende Funktion angewählt werden. Ist die jeweilige Funktion beendet, wird automatisch ins Hauptmenue zurückgekehrt.

In den beiden Kästen hinter "Funktion:" wird der Zustand der Flags für die Automatik und den Drucker angezeigt. "A" bedeutet Automatik an, "D" bedeutet Drucker ein.

Laden

Nach der Aufforderung, den Kassettenrecorder wiedergabebereit zu machen und deren Bestätigung, beginnt SAVETAPE das Programm zu lesen. Sobald der Header erkannt wurde, erscheinen in der oberen, rechten Bildschirmecke die zwei Sterne. Das File wird von da an Byte für Byte eingelesen und im Speicher abgelegt. Sollte eine Aufzeichnung am Ende kein vollständiges Byte mehr haben (bei geschützten Formaten), werden auch die restlichen Bits richtig geladen und gespeichert.

Während des Ladens läuft in der Bildschirmmitte der eingelesene Text durch. Weiterhin wird bei Systemfiles ein F gefolgt vom Filenamen, ein B für jeden Block und ein E für den Entrypointblock angezeigt.

Beispiel: COLMON

FCOLMONBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBE

Das Laden wird unter folgenden Bedingungen gestoppt:

1) Automatik Ein (Startbedingung)

sobald eine Pause in der Aufzeichnung auftritt.

2) Automatik Aus

sobald eine Pause auftritt und die <BREAK>-Taste gedrückt wird.

Will man den Ladevorgang mitten in einer Aufzeichnung beenden, kann man die Pause durch Anhalten des Kassettenrecorder erzeugen. Auch wenn der Header gesucht wird, also bevor die zwei Sterne erscheinen, ist es möglich, den Ladevorgang abubrechen. Hier ist nur erforderlich die <BREAK>-Taste zu drücken, der Kassettenrekorder braucht nicht angehalten zu werden.

Analyse der geladenen Aufzeichnung

Nach Beendigung des Ladevorgangs wird die geladenen Aufzeichnung analysiert. Dabei wird festgestellt, ob es sich um ein System- oder Basicfile handelt:

1) Systemfile:

Das Format für Systemfiles ist folgendes (siehe auch Colour-Genie-ROM-Listing):

Filename: 55H 6 Bytes Filename

Block: 3CH Länge Startadresse (LSB/MSB) Datenbytes
Prüfsumme

Entry: 78H Entrypoint (LSB/MSB)

Beispiel: COLMON (6000H-6FFFH)

Filename: 55H 43H 4FH 4CH 4DH 4FH 4EH = 55H COLMON

Block : 3CH 80H 00H 60H (128 Datenbytes) (Prüfsumme)
= 6000H-607FH
<30 ähnliche Blöcke>

Block : 3CH 80H 80H 6FH (128 Datenbytes) (Prüfsumme)
= 6F80H-6FFFH

Entry : 78H 00H 60H = 6000H

SAVETAPE erkennt automatisch das Systemformat und gibt in obigen Beispiel aus:

Systemfile: COLMON

Block : 6000H-6FFFH

Entry : 6000H

Bilden mehrere Blöcke des Systemfiles einen zusammenhängenden Speicherblock, wird nur dieser angezeigt (siehe Beispiel.).

Bei Blöcken zwischen 1 und 3 Bytes Länge wird zusätzlich der Inhalt des Blocks angezeigt. Da diese Blöcke oft Sprungvektoren sind, die überschrieben werden, ist diese Information besonders wertvoll. Ein Sprungbefehl der auf die Adresse 41E2H-41E4H geladen wird, bedeutet einen Autostart. Sie können also sofort die Startadresse eines Programmes mit Autostart ablesen, z.B.:

Block : 41E2H-41E4H : C3H E8H 41H

bedeutet, daß sobald der Entry im Systemfile erreicht, automatisch die Adresse 41E8H angesprungen wird.

Bei fehlerhaften Blöcken wird nur der Block des Systemfiles, der den Fehler enthielt, entsprechend markiert ausgedruckt:

Systemfile: COLMON

Block : 6000H-637FH

Block : 6380H-63FFH!!! Fehler !!!

Block : 6400H-6FFFH
Entry : 6000H

Vor jedem Block (55H, 3CH, 78H) können zusätzliche Bytes stehen, die vom System-Loader im ROM einfach überlesen werden. SAVETAPE zeigt deren Anzahl an. Bei Fehler erscheint auch häufig diese Anzeige:

Systemfile: COLMON
Block : 6000H-637FH
Block : 6380H-63FFH!!! Fehler !!!
zus. Bytes: 0073H
Block : 6480H-64FFH
Entry : 6000H

Auch nach dem Entry können bei geschützten Formaten zusätzliche Bytes auftreten.

2) Basicprogramme

Basicprogramme werden folgendermaßen aufgezeichnet:

Filename: 1 Byte
Basictext, wie im Speicher einschließlich der drei 0 am Ende.
(Siehe Colour-Genie-ROM-Listing)

Basicprogramme enthalten keine Prüfsumme, aber durch das Pointersystem (jede Zeile enthält einen Pointer auf die nächste Zeile) kann die geladene Aufzeichnung im gewissen Rahmen geprüft werden, wobei es unerheblich ist, wo das Basicprogramm vorher im Speicher stand.

Beispiel: Demoprogramm

Basicfile: D

Bei Fehlern im Pointersystem wird für jeden falschen Pointer die Meldung Fehler ausgegeben:

Basicfile: D
!!! Fehler !!!
!!! Fehler !!!

Bei beiden Filetypen kann eine weitere Fehlermeldung erscheinen:

File zu kurz.

Diese erscheint zum Beispiel, wenn ein Systemprogramm mitten im Block abbricht oder der Entry fehlt und bei einem Basicprogramm, wenn die drei 0 am Ende fehlen.

Bei Datenaufzeichnungen kann entweder nichts erscheinen oder es kann zufällig als einer der Filetypen erkannt werden. Dann erscheinen höchstwahrscheinlich zahlreiche

Fehlermeldungen.

Eine geladene Aufzeichnung seht bis zum nächsten Laden für Vergleiche und zum Aufzeichnen zur Verfügung

Die Analyse des Files wird bei eingeschaltetem Drucker (<D> im Hauptmenue) paralell zur Bildschirmausgabe auf den Drucker ausgegeben.

Vergleichen

Die in den Speicher geladenen Aufzeichnung kann mit einer weiteren verglichen werden. Abbruch wie beim Laden und automatisch beim 1. Ladefehler. Beim Vergleichen wird der eingelesene Text nicht auf dem Bildschirm angezeigt. Die Anzeige "Fname BBBBBBBBBBBE" erscheint, wenn das zu vergleichende File mit "L" geladen wurde, und nicht, wenn es mit "F" geladen wurde.

Schreiben

Das Format für Systemfiles auf Diskette ist folgendermaßen:

(Filename steht nur in Directory!)

Block : 01 Länge Startadresse (LSB/MSB) Datenbytes

Entry : 02 02 Einsprungadresse

Die Länge eines Blockes gilt einschließlich der Startadresse. Daher entspricht 00 = 254 Datenbytes, 01 = 255 und 02 = 256!)

Beispiel: COLMON

Block : 01H 82H 00H 60H (128 Datenbytes)

< 30 ähnliche Blöcke >

Block : 01H 82H 80H 6FH (128 Datenbytes)

Entry : 02H 02H 00H 60H

SAVETAPE setzt das oben beschriebene Kassettenformat in das Diskettenformat um. Viele Spielprogramme belegen Speicherbereiche, die das Disk-Basic auch benötigt. Ein direktes Laden auf diese Bereiche von Disk würde also zu einem Systemabsturz führen. Daher bietet SAVETAPE nach der Abfrage des Filenamen, welche mit einem korrekten Filenamen (Filename</Typ<.Password<:d>>>) beantwortet werden muß, die Möglichkeit das Programm zu Verschieben. Wenn diese Möglichkeit gewählt wird (durch <J>), wird das gesamte Programm (auch wenn es aus verschiedenen Speicherbereichen besteht) in einem Block startend mit der bei der Frage "neue Ladeadresse" hexdezimal angegebenen Adresse (<ENTER>

= 6000) auf Diskette gespeichert. Dahinter wird ein Zusatz gehängt, der beim Aufruf des Programmes mit CMD"S filename" dieses nach Abschalten des DOS wieder in die Speicherbereiche überträgt, in denen es ursprünglich lokalisiert war. Entrypoint und Autostart werden beachtet, so daß das Programm nach Laden und Verschieben ordnungsgemäß gestartet wird.

Ein Verschieben ist bei allen Programmen notwendig, die Blöcke haben, die unter 5C9AH laden, da das DOS den Speicherbereich 5800H- folgendermaßen nutzt:

5800H-5B77H : allgemeine Verwendung: Disk-Basic - DOS
nicht genau bekannt
5B78H-5C99H : Filebuffer "0" : FCB und Sektorbuffer
(5B9AH-5C99H) für CMD"S", CMD"L", SAVE, LOAD,
etc.
5C9AH-5DBBH : Filebuffer 1 für Disk-Basic
OPEN A\$,1,F\$
5DBCH-5EDDH : Filebuffer 2 dito
5EDEH-5FFFH : Filebuffer 3 dito
(bei mehr als 3 Files entsprechend weiter)
6000H- : Programmtext

Bestimmte Formate mit Preloader und/oder Bausrateumschaltung werden erkannt und entsprechend verarbeitet. Bei diesen Formaten (zu erkennen durch zahlreiche "zus. Bytes") ist ein Verschieben immer zu empfehlen.

Fehlertape laden

Die komplexen Analysevorgänge beim Laden machen das Laden mit SAVETAPE etwas kritischer als das normale Einladen mit CLOAD oder SYSTEM. Deshalb kann mit <F> eine weitere Laderoutine aufgerufen werden, die während des Laden nur den Text durchlaufen läßt, aber nicht "Fname BBBBBE" ausdrückt. Dadurch werden viele langwierige Operationen eingespart und das Laden ist erheblich unkritischer. Bei Ladeproblemen helfen im übrigen oft auch folgende Tips:

- 1) Höhen auf Maximum
- 2) Tonkopfjustage: Während der Wiedergabe auf maximale Höhenwiedergabe (deutlich zu hören) abstimmen. Nach erfolgreichem Laden wieder auf Ihre Standardeinstellung mit einer Ihrer alten Kassetten einstellen und dann die Kopien anfertigen.
- 3) Lautstärke: Bei manchen Aufzeichnungen hilft starkes übersteuern, bei anderen aber eher eine etwas niedrigere Wahl der Lautstärke (2).

Automatik

Die Automatik (Abbruch des Laden, wenn eine Pause auftritt) kann abgeschaltet werden, wenn eine Aufzeichnung mit Pausen geladen werden soll. Beim Starten des Programmes ist die Automatik ein. Durch Eingabe einer <A> im Hauptmenue kann sie ab- und wieder eingeschaltet werden. Der Zustand des Automatikflags kann im Hauptmenue jederzeit im linken Kasten abgelesen werden.

Drucker

Die Analyse eines geladenen Files kann wahlweise auf einen Drucker zusätzlich ausgedruckt werden. Wenn Sie dies wünschen, müssen Sie im Hauptmenue das <D> antippen. Mit der gleichen Taste können Sie den Drucker wieder abschalten. Der jeweilige Zustand kann im rechten Kasten abgelesen werden. Bitte achten Sie darauf, daß bei eingeschaltetem Drucker, auch wirklich ein Drucker angeschlossen und ON LINE ist. Sonst bleibt das Programm nach dem Laden stehen!

```

; Originaltexte von SAVETAPE
;
;
; Texte
;
TEXT1: DEFB      1CH
        DEFB      1FH
        DEFM      '      ***** SAVETAPE V1.0      *****'
        DEFB      CR
        DEFB      CR
        DEFM      'Kopiert Kassettenprogramme auf Diskette.'
        DEFB      CR
        DEFM      '(C) 1984 Luidger Roeckrath, 5100 Aachen'
        DEFB      CR
        DEFB      ' '
        DEFB      255
        DEFB      38
        DEFB      180
        DEFB      ' '
        DEFB      219
        DEFB      255
        DEFB      26
        DEFB      ' '
        DEFB      219
        DEFM      ' '
        DEFB      142
        DEFM      ' '
        DEFB      207
        DEFB      219
        DEFM      ' Funktionen:'
        DEFB      255
        DEFB      13
        DEFB      ' '
        DEFB      219
        DEFM      ' '
BA:      DEFM      'A '
        DEFB      142
        DEFM      ' '
DA:      DEFM      ' '
        DEFB      207
        DEFB      219
        DEFB      255
        DEFB      26
        DEFB      ' '
        DEFB      219
        DEFM      ' '
        DEFB      142
        DEFM      ' '
        DEFB      207
        DEFB      219
        DEFB      255
        DEFB      38
        DEFB      181
        DEFB      207
        DEFB      219
        DEFB      255
        DEFB      38
        DEFB      ' '
        DEFB      207

```

```
DEFB      219
DEFM      '  <L> a d e n '
DEFB      255
DEFB      25
DEFB      ' ,
DEFB      207
DEFB      219
DEFB      255
DEFB      38
DEFB      ' ,
DEFB      207
DEFB      219
DEFM      '  <U> e r g l e i c h e n '
DEFB      255
DEFB      13
DEFB      ' ,
DEFB      207
DEFB      219
DEFB      255
DEFB      38
DEFB      ' ,
DEFB      207
DEFB      219
DEFM      '  <S> c h r e i b e n a u f D i s k '
DEFB      207
DEFB      219
DEFB      255
DEFB      38
DEFB      ' ,
DEFB      207
DEFB      219
DEFM      '  <F> e h l e r t a p e   l a d e n   '
DEFB      207
DEFB      219
DEFB      255
DEFB      38
DEFB      ' ,
DEFB      207
DEFB      219
DEFM      '  <A> u t o m a t i k ( E i n / A u s ) '
DEFB      207
DEFB      219
DEFB      255
DEFB      38
DEFB      ' ,
DEFB      207
DEFB      219
DEFM      '  <D> r u c k e r ( E i n / A u s )   '
DEFB      207
DEFB      219
DEFB      255
DEFB      38
DEFB      ' ,
DEFB      207
DEFB      ' ,
DEFB      255
DEFB      38
DEFB      181
DEFB      ' ,
DEFB      0
```

```

TEXT2:  DEFB      1CH
        DEFB      1FH
        DEFM      'Kassettenrecorder auf Wiedergabe'
        DEFB      CR
        DEFB      CR
        DEFB      0
TEXT3:  DEFB      CR
        DEFB      CR
        DEFM      'Laenge des Files:'
        DEFB      0
TEXT4:  DEFB      1CH
        DEFB      1FH
        DEFM      'Filename'
        DEFB      0
TEXT5B: DEFM      'Verschieben'
        DEFB      0
TEXT5A: DEFM      'Neue Ladeadresse'
        DEFB      0
TEXT6:  DEFM      'L a d e n . . . '
TEXT6A: DEFB      CR
        DEFB      CR
        DEFB      0
TEXT7:  DEFM      'S c h r e i b e n . . . '
        DEFB      CR
        DEFB      CR
        DEFB      0
TEXT8:  DEFM      'V e r g l e i c h e n . . . '
        DEFB      CR
        DEFB      CR
        DEFB      0
TEXT0:  DEFM      'Weiter. Bitte <ENTER> druecken'
        DEFB      CR
        DEFB      CR
        DEFB      0
TEXT10: DEFB      CR
        DEFM      'Systemfile: '
        DEFB      0
TEXT11: DEFB      CR
        DEFM      'Block      : '
        DEFB      0
TEXT18: DEFB      CR
        DEFM      'zus. Bytes: '
        DEFB      0
TEXT12: DEFB      CR
        DEFM      'Entry      : '
        DEFB      0
TEXT13: DEFM      '!!! Fehler !!!'
        DEFB      0
TEXT14: DEFM      'Basicfile : '
        DEFB      0
TEXT15: DEFM      'Kein '
TEXT5:  DEFM      'Vergleichsfehler'
        DEFB      CR
        DEFB      CR
        DEFB      0
TEXT16: DEFM      ' ? '
        DEFB      0
TEXT17: DEFM      ' : Disk-Fehler'
        DEFB      CR
        DEFB      CR

```

```
      DEFB      0
TEXT19: DEFB      CR
      DEFM      'File zu kurz.'
      DEFB      CR
      DEFB      CR
      DEFB      0
TEXT20: DEFB      ' : '
      DEFB      0
      ;
      END
```

```

00100      ; SAVETAPE Version 1.0 vom 18.4.84
00110      ;
00120      ; Kopiert Colour-Genie Kassetten auf Diskette
00130      ;
00140      ; (c) Luidger Röckrath
00150      ;   Noppiusstr. 19;
00160      ;   5100 Aachen
00170      ;   (0241)34962
00180      ;
00190      ;
00200      ;
000D      00210 CR    EQU    0DH
00220      ;
00230      ;
00240      ;
00250      ;ASEG
4800      00260      ORG    4800H
00270      ;
4800 F3    00280 START DI
4801 31EF57 00290      LD    SP,57EFH
4804 3E69    00300      LD    A,105
4806 321243 00310      LD    (4312H),A
4809 3E03    00320      LD    A,3
480B CDB03F 00330      CALL 3FB0H
480E 21C6F4 00340      LD    HL,TEXT1
4811 CD00F4 00350      CALL PRINT
4814 CD2B00 00360 ST1    CALL INCH
4817 FE61    00370      CP    'a'
4819 3802    00380      JR    C,ST2
481B D620    00390      SUB   'a'-'A'
481D FE4C    00400 ST2    CP    'L'
481F 281C    00410      JR    Z,LADEN
4821 FE53    00420      CP    'S'
4823 CAF94A 00430      JP    Z,SCHREI
4826 FE56    00440      CP    'V'
4828 CA9351 00450      JP    Z,VERGLE
482B FE41    00460      CP    'A'
482D CAD34A 00470      JP    Z,AUTOM
4830 FE44    00480      CP    'D'
4832 CAE64A 00490      JP    Z,DREA
4835 FE46    00500      CP    'F'
4837 CA3C48 00510      JP    Z,FTAPEL
483A 1808    00520      JR    ST1
00530      ;
00540      ; Fehlerhaftes Tape laden
00550      ;
483C 3E      00560 FTAPEL DEFB 3EH      ; = LD A,...
483D AF      00570 LADEN  XOR    A
483E 3295F5 00580      LD    (FTF),A
4841 21E8F4 00590      LD    HL,TEXT2
4844 CD00F4 00600      CALL PRINT
4847 210000 00610      LD    HL,0
484A 2289F5 00620      LD    (LSTREC),HL
484D 2287F5 00630      LD    (LSTRAC),HL
4850 CDEB51 00640      CALL WAIT
4853 211EF5 00650      LD    HL,TEXT6
4856 CD00F4 00660      CALL PRINT
4859 117F46 00670      LD    DE,467FH
485C 211CCB 00680      LD    HL,0CB1CH
485F 22C851 00690      LD    (READY),HL

```

| | | | | | |
|------|----------|-------|--------|------|--------------|
| 4862 | FD2A2040 | 00700 | LAD | LD | IY, (4020H) |
| 4866 | 3E55 | 00710 | | LD | A, 'U' |
| 4868 | 328648 | 00720 | | LD | (L1110+1), A |
| 486B | 21775B | 00730 | | LD | HL, BUFFER-1 |
| 486E | CD3C52 | 00740 | | CALL | READS |
| 4871 | DA4E4A | 00750 | | JP | C, L110 |
| 4874 | 3A95F5 | 00760 | | LD | A, (FTF) |
| 4877 | A7 | 00770 | | AND | A |
| 4879 | 207D | 00780 | | JR | NZ, L1116 |
| 487A | CDC851 | 00790 | L1 | CALL | READY |
| 487D | FE3C | 00800 | | CP | 3CH |
| 487F | 280A | 00810 | | JR | Z, L1111 |
| 4881 | FE78 | 00820 | | CP | 78H |
| 4883 | 2861 | 00830 | | JR | Z, L1115 |
| 4885 | FE55 | 00840 | L1110 | CP | 55H |
| 4887 | 2844 | 00850 | | JR | Z, L1114 |
| 4889 | 18EF | 00860 | | JR | L1 |
| 488B | FD360042 | 00870 | L1111 | LD | (IY+0), 'B' |
| 488F | FD23 | 00880 | | INC | IY |
| 4891 | AF | 00890 | | XOR | A |
| 4892 | 3294F5 | 00900 | | LD | (CF), A |
| 4895 | CDC851 | 00910 | | CALL | READY |
| 4898 | 47 | 00920 | | LD | B, A |
| 4899 | CDC851 | 00930 | | CALL | READY |
| 489C | D9 | 00940 | | EXX | |
| 489D | 6F | 00950 | | LD | L, A |
| 489E | D9 | 00960 | | EXX | |
| 489F | CDC851 | 00970 | | CALL | READY |
| 48A2 | D9 | 00980 | | EXX | |
| 48A3 | 67 | 00990 | | LD | H, A |
| 48A4 | D9 | 01000 | | EXX | |
| 48A5 | CDC851 | 01010 | L1112 | CALL | READY |
| 48A8 | D9 | 01020 | | EXX | |
| 48A9 | 7C | 01030 | | LD | A, H |
| 48AA | FE43 | 01040 | | CP | 43H |
| 48AC | 7D | 01050 | | LD | A, L |
| 48AD | 23 | 01060 | | INC | HL |
| 48AE | D9 | 01070 | | EXX | |
| 48AF | 2010 | 01080 | | JR | NZ, L1113 |
| 48B1 | FE12 | 01090 | | CP | 12H |
| 48B3 | 200C | 01100 | | JR | NZ, L1113 |
| 48B5 | 7E | 01110 | | LD | A, (HL) |
| 48B6 | 321243 | 01120 | | LD | (4312H), A |
| 48B9 | 3669 | 01130 | | LD | (HL), 105 |
| 48BB | ED44 | 01140 | | NEG | |
| 48BD | 86 | 01150 | | ADD | A, (HL) |
| 48BE | 3294F5 | 01160 | | LD | (CF), A |
| 48C1 | 10E2 | 01170 | L1113 | DJNZ | L1112 |
| 48C3 | CDC851 | 01180 | | CALL | READY |
| 48C6 | 3A94F5 | 01190 | | LD | A, (CF) |
| 48C9 | 86 | 01200 | | ADD | A, (HL) |
| 48CA | 77 | 01210 | | LD | (HL), A |
| 48CB | 18AD | 01220 | | JR | L1 |
| 48CD | FD360046 | 01230 | L1114 | LD | (IY+0), 'F' |
| 48D1 | FD23 | 01240 | | INC | IY |
| 48D3 | 0606 | 01250 | | LD | B, 6 |
| 48D5 | CDC851 | 01260 | L11114 | CALL | READY |
| 48D8 | FD7700 | 01270 | | LD | (IY+0), A |
| 48DB | FD23 | 01280 | | INC | IY |
| 48DD | 10F6 | 01290 | | DJNZ | L11114 |

| | | | |
|---------------|-------------|------|--------------|
| 48DF 3E3C | 01300 | LD | A, 3CH |
| 48E1 328648 | 01310 | LD | (L1110+1), A |
| 48E4 1894 | 01320 | JR | L1 |
| 48E6 FD360045 | 01330 L1115 | LD | (IY+0), 'E' |
| 48EA FD23 | 01340 | INC | IY |
| 48EC CDC851 | 01350 | CALL | READY |
| 48EF CDC851 | 01360 | CALL | READY |
| 48F2 3E69 | 01370 | LD | A, 105 |
| 48F4 321243 | 01380 | LD | (4312H), A |
| 48F7 CDC851 | 01390 L1116 | CALL | READY |
| 48FA 18FB | 01400 | JR | L1116 |
| 48FC 3E00 | 01410 L22 | LD | A, 0 |
| 48FE BF | 01420 | ADC | A, A |
| 48FF 3296F5 | 01430 | LD | (AFF), A |
| 4902 FD222040 | 01440 | LD | (4020H), IY |
| 4906 3A92F5 | 01450 | LD | A, (DF) |
| 4909 3293F5 | 01460 | LD | (DFLAG), A |
| 490C 11785B | 01470 | LD | DE, BUFFER |
| 490F A7 | 01480 | AND | A |
| 4910 ED52 | 01490 | SBC | HL, DE |
| 4912 E5 | 01500 | PUSH | HL |
| 4913 21F6F4 | 01510 | LD | HL, TEXT3 |
| 4916 CD00F4 | 01520 | CALL | PRINT |
| 4919 E1 | 01530 | POP | HL |
| 491A CD54F4 | 01540 | CALL | PRHEX |
| 491D CDB94A | 01550 | CALL | CRCR |
| 4920 2285F5 | 01560 | LD | (FLEN), HL |
| 4923 118046 | 01570 | LD | DE, 4680H |
| 4926 0680 | 01580 | LD | B, 128 |
| 4928 3E20 | 01590 | LD | A, ' ' |
| 492A 12 | 01600 L113 | LD | (DE), A |
| 492B 13 | 01610 | INC | DE |
| 492C 10FC | 01620 | DJNZ | L113 |
| 492E 11785B | 01630 | LD | DE, BUFFER |
| 4931 2A85F5 | 01640 | LD | HL, (FLEN) |
| 4934 19 | 01650 | ADD | HL, DE |
| 4935 2B | 01660 | DEC | HL |
| 4936 3A96F5 | 01670 | LD | A, (AFF) |
| 4939 A7 | 01680 | AND | A |
| 493A 2801 | 01690 | JR | Z, L114 |
| 493C 2B | 01700 | DEC | HL |
| 493D CDA54A | 01710 L114 | CALL | NULL |
| 4940 CA584A | 01720 | JP | Z, L100 |
| 4943 1A | 01730 | LD | A, (DE) |
| 4944 FE55 | 01740 | CP | 'U' |
| 4946 2806 | 01750 | JR | Z, L115 |
| 4948 CDA54A | 01760 | CALL | NULL |
| 494B CA584A | 01770 | JP | Z, L100 |
| 494E 21FFFF | 01780 L115 | LD | HL, 0FFFFH |
| 4951 1A | 01790 L112 | LD | A, (DE) |
| 4952 CDACF4 | 01800 | CALL | ENDE |
| 4955 23 | 01810 | INC | HL |
| 4956 13 | 01820 | INC | DE |
| 4957 FE55 | 01830 | CP | 'U' |
| 4959 20F6 | 01840 | JR | NZ, L112 |
| 495B 7C | 01850 | LD | A, H |
| 495C B5 | 01860 | OR | L |
| 495D E5 | 01870 | PUSH | HL |
| 495E F5 | 01880 | PUSH | AF |
| 495F 2144F5 | 01890 | LD | HL, TEXT18 |

| | | | | |
|------|----------|------------|------|------------|
| 4962 | C400F4 | 01900 | CALL | NZ,PRINT |
| 4965 | F1 | 01910 | POP | AF |
| 4966 | E1 | 01920 | POP | HL |
| 4967 | CDC14A | 01930 | CALL | MZUS |
| 496A | C454F4 | 01940 | CALL | NZ,PRHEX |
| 496D | 2136F5 | 01950 | LD | HL,TEXT10 |
| 4970 | CD00F4 | 01960 | CALL | PRINT |
| 4973 | 0606 | 01970 | LD | B,6 |
| 4975 | 1A | 01980 L11 | LD | A,(DE) |
| 4976 | CD28F4 | 01990 | CALL | OUTCH |
| 4979 | 13 | 02000 | INC | DE |
| 497A | 10F9 | 02010 | DJNZ | L11 |
| 497C | 21FFFF | 02020 L12 | LD | HL,OFFFHH |
| 497F | 1A | 02030 L121 | LD | A,(DE) |
| 4980 | CDACF4 | 02040 | CALL | ENDE |
| 4983 | 13 | 02050 | INC | DE |
| 4984 | 23 | 02060 | INC | HL |
| 4985 | FE78 | 02070 | CP | 78H |
| 4987 | 2804 | 02080 | JR | Z,L122 |
| 4989 | FE3C | 02090 | CP | 3CH |
| 498B | 20F2 | 02100 | JR | NZ,L121 |
| 498D | F5 | 02110 L122 | PUSH | AF |
| 498E | 7C | 02120 | LD | A,H |
| 498F | B5 | 02130 | OR | L |
| 4990 | 280E | 02140 | JR | Z,L123 |
| 4992 | CDC14A | 02150 | CALL | MZUS |
| 4995 | E5 | 02160 | PUSH | HL |
| 4996 | 2144F5 | 02170 | LD | HL,TEXT18 |
| 4999 | CD00F4 | 02180 | CALL | PRINT |
| 499C | E1 | 02190 | POP | HL |
| 499D | CD54F4 | 02200 | CALL | PRHEX |
| 49A0 | F1 | 02210 L123 | POP | AF |
| 49A1 | FE78 | 02220 | CP | 78H |
| 49A3 | 2873 | 02230 | JR | Z,L15 |
| 49A5 | 1A | 02240 | LD | A,(DE) |
| 49A6 | 4F | 02250 | LD | C,A |
| 49A7 | 13 | 02260 | INC | DE |
| 49A8 | 1A | 02270 | LD | A,(DE) |
| 49A9 | 6F | 02280 | LD | L,A |
| 49AA | 13 | 02290 | INC | DE |
| 49AB | 1A | 02300 | LD | A,(DE) |
| 49AC | 67 | 02310 | LD | H,A |
| 49AD | 13 | 02320 | INC | DE |
| 49AE | 85 | 02330 | ADD | A,L |
| 49AF | 47 | 02340 | LD | B,A |
| 49B0 | C5 | 02350 | PUSH | BC |
| 49B1 | ED538BF5 | 02360 | LD | (LSTAD),DE |
| 49B5 | 1A | 02370 L14 | LD | A,(DE) |
| 49B6 | CDACF4 | 02380 | CALL | ENDE |
| 49B9 | 13 | 02390 | INC | DE |
| 49BA | 80 | 02400 | ADD | A,B |
| 49BB | 47 | 02410 | LD | B,A |
| 49BC | 0D | 02420 | DEC | C |
| 49BD | 20F6 | 02430 | JR | NZ,L14 |
| 49BF | 1A | 02440 | LD | A,(DE) |
| 49C0 | 13 | 02450 | INC | DE |
| 49C1 | 90 | 02460 | SUB | B |
| 49C2 | 280C | 02470 | JR | Z,L18 |
| 49C4 | E5 | 02480 | PUSH | HL |
| 49C5 | CD5CF4 | 02490 | CALL | PRBL |

```

49C8 E1      02500      POP      HL
49C9 010000   02510      LD       BC,0
49CC ED4389F5 02520      LD       (LSTREC),BC
49D0 328FF5   02530 L18  LD       (CSF),A
49D3 C1       02540      POP      BC
49D4 E5       02550      PUSH     HL
49D5 D5       02560      PUSH     DE
49D6 ED5B89F5 02570      LD       DE, (LSTREC)
49DA 7B       02580      LD       A,E
49DB B2       02590      OR       D
49DC 2003     02600      JR       NZ,L16
49DE 3C       02610      INC      A
49DF 1808     02620      JR       L17
49E1 37       02630 L16  SCF
49E2 ED52     02640      SBC      HL,DE
49E4 F5       02650      PUSH     AF
49E5 C45CF4   02660      CALL    NZ,PRBL
49E8 F1       02670      POP      AF
49E9 D1       02680 L17  POP      DE
49EA E1       02690      POP      HL
49EB 280B     02700      JR       Z,L13
49ED 2287F5   02710      LD       (LSTRAC),HL
49F0 E5       02720      PUSH     HL
49F1 2A8BF5   02730      LD       HL, (LSTAD)
49F4 228DF5   02740      LD       (LSTAD1),HL
49F7 E1       02750      POP      HL
49F8 0600     02760 L13  LD       B,0
49FA 0D       02770      DEC      C
49FB 09       02780      ADD      HL,BC
49FC 2289F5   02790      LD       (LSTREC),HL
49FF 3A8FF5   02800      LD       A, (CSF)
4A02 A7       02810      AND      A
4A03 CA7C49   02820      JP       Z,L12
4A06 CD5CF4   02830      CALL    PRBL
4A09 210000   02840      LD       HL,0
4A0C 2289F5   02850      LD       (LSTREC),HL
4A0F 2152F5   02860      LD       HL,TEXT13
4A12 CD00F4   02870      CALL    PRINT
4A15 C37C49   02880      JP       L12
               02890      ;
4A18 CD5CF4   02900 L15  CALL    PRBL
4A1B 214BF5   02910      LD       HL,TEXT12
4A1E CD00F4   02920      CALL    PRINT
4A21 1A       02930      LD       A, (DE)
4A22 6F       02940      LD       L,A
4A23 13       02950      INC      DE
4A24 1A       02960      LD       A, (DE)
4A25 67       02970      LD       H,A
4A26 CD54F4   02980      CALL    PRHEX
4A29 13       02990      INC      DE
4A2A ED5398F5 03000      LD       (ENDSYS),DE
4A2E 21785B   03010      LD       HL,BUFFER
4A31 ED4B85F5 03020      LD       BC, (FLEN)
4A35 09       03030      ADD      HL,BC
4A36 A7       03040      AND      A
4A37 ED52     03050      SBC      HL,DE
4A39 7D       03060      LD       A,L
4A3A B4       03070      OR       H
4A3B 280E     03080      JR       Z,L111
4A3D E5       03090      PUSH     HL

```

```

4A3E 2144F5 03100 LD HL,TE,T16
4A41 CD00F4 03110 CALL PRINT
4A44 E1 03120 POP HL
4A45 CDC14A 03130 CALL MZUS
4A48 CD54F4 03140 CALL PFHEX
4A4B CDB94A 03150 L111 CALL CRCP
4A4E AF 03160 L110 POP A
4A4F 3293F5 03170 LD (DFLAG),A
4A52 CDEB51 03180 WSTART CALL WAIT
4A55 C30048 03190 JP START
03200 ;
4A58 2156F5 03210 L100 LD HL,TE,T14
4A5B CD00F4 03220 CALL PRINT
4A5E 1A 03230 LD A,(DE)
4A5F CD28F4 03240 CALL DUTCH
4A62 13 03250 INC DE
4A63 CDB94A 03260 CALL CRCP
4A66 EB 03270 EX DE,HL
4A67 5E 03280 L105 LD E,(HL)
4A69 23 03290 INC HL
4A69 56 03300 LD D,(HL)
4A6A 23 03310 INC HL
4A6B 23 03320 INC HL
4A6C 23 03330 INC HL
4A6D 7E 03340 L101 LD A,(HL)
4A6E CDA6F4 03350 CALL ENDE1
4A71 23 03360 INC HL
4A72 A7 03370 AND A
4A73 20F8 03380 JR NZ,L101
4A75 CDA6F4 03390 L102 CALL ENDE1
4A78 D5 03400 PUSH DE
4A79 5E 03410 LD E,(HL)
4A7A 23 03420 INC HL
4A7B 56 03430 LD D,(HL)
4A7C 7A 03440 LD A,D
4A7D B3 03450 OR E
4A7E E3 03460 EX (SP),HL
4A7F EB 03470 EX DE,HL
4A80 C1 03480 POP BC
4A81 CA4E4A 03490 JP Z,L110
4A84 E5 03500 PUSH HL
4A85 ED52 03510 SBC HL,DE
4A87 09 03520 ADD HL,BC
4A88 D1 03530 POP DE
4A89 2B 03540 DEC HL
4A8A 2B 03550 DEC HL
4A8B 7E 03560 LD A,(HL)
4A8C 23 03570 INC HL
4A8D A7 03580 AND A
4A8E 28E5 03590 JR Z,L102
4A90 E5 03600 PUSH HL
4A91 2152F5 03610 LD HL,TEXT13
4A94 CD00F4 03620 CALL PRINT
4A97 E1 03630 POP HL
4A98 CDBC4A 03640 CALL CR1
4A9B 7E 03650 L103 LD A,(HL)
4A9C CDA6F4 03660 CALL ENDE1
4A9F 23 03670 INC HL
4AA0 A7 03680 AND A
4AA1 20F8 03690 JR NZ,L103

```

```

4AA3 18C2      03700      JR      L105
                03710 ;
4AA5 7E        03720 NULL  LD      A, (HL)
4AA6 2B        03730      DEC     HL
4AA7 B6        03740      OR      (HL)
4AA8 2B        03750      DEC     HL
4AA9 B6        03760      OR      (HL)
4AAA 23        03770      INC     HL
4AAB C0        03780      RET     NZ
4AAC 2B        03790      DEC     HL
4AAD 2B        03800      DEC     HL
4AAE 7E        03810      LD      A, (HL)
4AAF FE7B      03820      CP      7BH
4AB1 23        03830      INC     HL
4AB2 23        03840      INC     HL
4AB3 2802      03850      JR      Z, N1
4AB5 AF        03860      XOR     A
4AB6 C9        03870      RET
4AB7 A7        03880 N1    AND     A
4AB8 C9        03890      RET
                03900      ;
4AB9 CDBC4A    03910 CRCR  CALL    CR1
4ABC 3E0D      03920 CR1  LD      A, CR
4ABE C328F4    03930      JP      DUTCH
                03940      ;
4AC1 F5        03950 MZUS  PUSH    AF
4AC2 7C        03960      LD      A, H
4AC3 A7        03970      AND     A
4AC4 3297F5    03980      LD      (MZ), A
4AC7 2008      03990      JR      NZ, MZ1
4AC9 7D        04000      LD      A, L
4ACA FE0A      04010      CP      10
4ACC 3803      04020      JR      C, MZ1
4ACE 3297F5    04030      LD      (MZ), A
4AD1 F1        04040 MZ1  POP     AF
4AD2 C9        04050      RET
                04060      ;
                04070      ; Automatik (Ein/Aus)
                04080      ;
4AD3 3A90F5    04090 AUTOM LD      A, (BF)
4AD6 EEFF      04100      XOR     255
4AD8 3290F5    04110      LD      (BF), A
4ADB 3AD1F4    04120      LD      A, (BA)
4ADE EE61      04130      XOR     'a' ; = ' ' XOR 'A'
4AE0 32D1F4    04140      LD      (BA), A
4AE3 C30048    04150      JP      START
                04160      ;
                04170      ; Drucker (Ein/Aus)
                04180      ;
4AE6 3A92F5    04190 DREA LD      A, (DF)
4AE9 EEFF      04200      XOR     255
4AEB 3292F5    04210      LD      (DF), A
4AEE 3AD3F4    04220      LD      A, (DA)
4AF1 EE64      04230      XOR     'd' ; = ' ' XOR 'D'
4AF3 32D3F4    04240      LD      (DA), A
4AF6 C30048    04250      JP      START
                04260      ;
                04270      ; Schreiben
                04280      ;
4AF9 21FEF4    04290 SCHREI LD      HL, TEXT4

```

| | | | |
|---------------|------------|------|---------------|
| 4AFC 11A752 | 04300 | LD | DE,FCB |
| 4AFF 0617 | 04310 | LD | B,23 |
| 4B01 CD8351 | 04320 | CALL | INPUT |
| 4B04 EB | 04330 | EX | DE,HL |
| 4B05 21C952 | 04340 | LD | HL,DBUF |
| 4B08 0600 | 04350 | LD | B,0 |
| 4B0A CD20CE | 04360 | CALL | INIT |
| 4B0D C28752 | 04370 | JP | NZ,DERROR |
| 4B10 2109F5 | 04380 S001 | LD | HL,TEXT5B |
| 4B13 11C952 | 04390 | LD | DE,DBUF |
| 4B16 0601 | 04400 | LD | B,1 |
| 4B18 CD8351 | 04410 | CALL | INPUT |
| 4B1B 7E | 04420 | LD | A,(HL) |
| 4B1C E6DF | 04430 | AND | 0DFH |
| 4B1E FE4E | 04440 | CP | 'N' |
| 4B20 2822 | 04450 | JR | Z,S002 |
| 4B22 FE4A | 04460 | CP | 'J' |
| 4B24 20EA | 04470 | JR | NZ,S001 |
| 4B26 2115F5 | 04480 | LD | HL,TEXT5A |
| 4B29 11C952 | 04490 | LD | DE,DBUF |
| 4B2C 0604 | 04500 | LD | B,4 |
| 4B2E CD8351 | 04510 | CALL | INPUT |
| 4B31 7E | 04520 | LD | A,(HL) |
| 4B32 FE0D | 04530 | CP | CR |
| 4B34 110060 | 04540 | LD | DE,6000H |
| 4B37 2806 | 04550 | JR | Z,S003 |
| 4B39 110000 | 04560 | LD | DE,0 |
| 4B3C CDFE34 | 04570 | CALL | 34FEH |
| 4B3F ED539BF5 | 04580 S003 | LD | (VERADR),DE |
| 4B43 AF | 04590 | XOR | A |
| 4B44 329AF5 | 04600 S002 | LD | (VF),A |
| 4B47 2123F5 | 04610 | LD | HL,TEXT7 |
| 4B4A CD00F4 | 04620 | CALL | PRINT |
| 4B4D 3A97F5 | 04630 | LD | A,(MZ) |
| 4B50 A7 | 04640 | AND | A |
| 4B51 21785B | 04650 | LD | HL,BUFFER |
| 4B54 282A | 04660 | JR | Z,S01 |
| 4B56 2A98F5 | 04670 | LD | HL,(ENDSYS) |
| 4B59 E5 | 04680 | PUSH | HL |
| 4B5A 11F15B | 04690 | LD | DE,BUFFER+79H |
| 4B5D 0E32 | 04700 | LD | C,50 |
| 4B5F 7E | 04710 S02 | LD | A,(HL) |
| 4B60 FE7B | 04720 | CP | 7BH |
| 4B62 281B | 04730 | JR | Z,S05 |
| 4B64 FE3C | 04740 | CP | 3CH |
| 4B66 23 | 04750 | INC | HL |
| 4B67 20F6 | 04760 | JR | NZ,S02 |
| 4B69 46 | 04770 | LD | B,(HL) |
| 4B6A 23 | 04780 | INC | HL |
| 4B6B 23 | 04790 | INC | HL |
| 4B6C 23 | 04800 | INC | HL |
| 4B6D 1A | 04810 S03 | LD | A,(DE) |
| 4B6E AE | 04820 | XOR | (HL) |
| 4B6F 77 | 04830 | LD | (HL),A |
| 4B70 13 | 04840 | INC | DE |
| 4B71 23 | 04850 | INC | HL |
| 4B72 0D | 04860 | DEC | C |
| 4B73 2005 | 04870 | JR | NZ,S04 |
| 4B75 0E32 | 04880 | LD | C,50 |
| 4B77 11F15B | 04890 | LD | DE,BUFFER+79H |

```

4B7A 10F1      04900 S04      DJNZ      S03
4B7C 23        04910          INC        HL
4B7D 18E0      04920          JR          S02
4B7F E1        04930 S05      POP        HL
4B80 ED4B9BF5  04940 S01      LD         BC, (VERADR)
4B84 ED43B44C  04950          LD         (VERAD2+1), BC
4B88 110000    04960          LD         DE, 0
4B8B ED539DF5  04970          LD         (AE), DE
4B8F 11CF4C    04980          LD         DE, BTAB
4B92 7E        04990 S1      LD         A, (HL)
4B93 FE3C      05000          CP         3CH
4B95 23        05010          INC        HL
4B96 20FA      05020          JR          NZ, S1
4B98 3E01      05030          LD         A, 1
4B9A CD7E52    05040          CALL      WRB
4B9D 7E        05050          LD         A, (HL)
4B9E F5        05060          PUSH     AF
4B9F C602      05070          ADD        A, 2
4BA1 CD7E52    05080          CALL      WRB
4BA4 23        05090          INC        HL
4BA5 3A9AF5    05100          LD         A, (VF)
4BA8 A7        05110          AND        A
4BA9 2B1B      05120          JR          Z, S004
4BAB 4E        05130          LD         C, (HL)
4BAC 23        05140          INC        HL
4BAD 46        05150          LD         B, (HL)
4BAE 2B        05160          DEC        HL
4BAF E5        05170          PUSH     HL
4BB0 21E241    05180          LD         HL, 41E2H
4BB3 ED42      05190          SBC        HL, BC
4BB5 E1        05200          POP        HL
4BB6 200E      05210          JR          NZ, S004
4BB8 E5        05220          PUSH     HL
4BB9 23        05230          INC        HL
4BBA 23        05240          INC        HL
4BBB 23        05250          INC        HL
4BBC C5        05260          PUSH     BC
4BBD 4E        05270          LD         C, (HL)
4BBE 23        05280          INC        HL
4BBF 46        05290          LD         B, (HL)
4BC0 ED439DF5  05300          LD         (AE), BC
4BC4 C1        05310          POP        BC
4BC5 E1        05320          POP        HL
4BC6 79        05330 S004     LD         A, C
4BC7 CD7E52    05340          CALL      WRB
4BCA 78        05350          LD         A, B
4BCB CD7E52    05360          CALL      WRB
4BCE E5        05370          PUSH     HL
4BCF D5        05380          PUSH     DE
4BD0 216F4D    05390          LD         HL, BLOCK1
4BD3 A7        05400          AND        A
4BD4 ED52      05410          SBC        HL, DE
4BD6 2B12      05420          JR          Z, S111
4BD8 110401    05430          LD         DE, 260
4BDB A7        05440          AND        A
4BDC ED5A      05450          ADC        HL, DE
4BDE 280A      05460          JR          Z, S111
4BE0 A7        05470          AND        A
4BE1 ED5A      05480          ADC        HL, DE
4BE3 2805      05490          JR          Z, S111

```

| | | | |
|---------------|------------|------|-------------------|
| 4BE5 A7 | 05500 | AND | A |
| 4BE6 ED5A | 05510 | ADC | HL, DE |
| 4BE8 2006 | 05520 | JR | NZ, S11 |
| 4BEA D1 | 05530 S111 | POP | DE |
| 4BEB 13 | 05540 | INC | DE |
| 4BEC 13 | 05550 | INC | DE |
| 4BED 13 | 05560 | INC | DE |
| 4BEE 13 | 05570 | INC | DE |
| 4BEF D5 | 05580 | PUSH | DE |
| 4BF0 D1 | 05590 S11 | POP | DE |
| 4BF1 E1 | 05600 | POP | HL |
| 4BF2 7E | 05610 | LD | A, (HL) |
| 4BF3 12 | 05620 | LD | (DE), A |
| 4BF4 23 | 05630 | INC | HL |
| 4BF5 13 | 05640 | INC | DE |
| 4BF6 7E | 05650 | LD | A, (HL) |
| 4BF7 12 | 05660 | LD | (DE), A |
| 4BF8 23 | 05670 | INC | HL |
| 4BF9 13 | 05680 | INC | DE |
| 4BFA F1 | 05690 | POP | AF |
| 4BFB 12 | 05700 | LD | (DE), A |
| 4BFC F5 | 05710 | PUSH | AF |
| 4BFD FE01 | 05720 | CP | 1 |
| 4BFF 3E00 | 05730 | LD | A, 0 |
| 4C01 3F | 05740 | ADC | A, A |
| 4C02 13 | 05750 | INC | DE |
| 4C03 12 | 05760 | LD | (DE), A |
| 4C04 13 | 05770 | INC | DE |
| 4C05 F1 | 05780 | POP | AF |
| 4C06 D5 | 05790 | PUSH | DE |
| 4C07 57 | 05800 | LD | D, A |
| 4C08 7E | 05810 S2 | LD | A, (HL) |
| 4C09 CD7E52 | 05820 | CALL | WRB |
| 4C0C 23 | 05830 | INC | HL |
| 4C0D 03 | 05840 | INC | BC |
| 4C0E 15 | 05850 | DEC | D |
| 4C0F 20F7 | 05860 | JR | NZ, S2 |
| 4C11 23 | 05870 | INC | HL |
| 4C12 D1 | 05880 | POP | DE |
| 4C13 7E | 05890 | LD | A, (HL) |
| 4C14 FE78 | 05900 | CP | 78H |
| 4C16 C2924B | 05910 | JP | NZ, S1 |
| 4C19 AF | 05920 | XOR | A |
| 4C1A 13 | 05930 | INC | DE |
| 4C1B 13 | 05940 | INC | DE |
| 4C1C 12 | 05950 | LD | (DE), A |
| 4C1D 13 | 05960 | INC | DE |
| 4C1E 12 | 05970 | LD | (DE), A |
| 4C1F 23 | 05980 | INC | HL |
| 4C20 7E | 05990 | LD | A, (HL) |
| 4C21 32CD4C | 06000 | LD | (ENTRY+1), A |
| 4C24 23 | 06010 | INC | HL |
| 4C25 7E | 06020 | LD | A, (HL) |
| 4C26 32CE4C | 06030 | LD | (ENTRY+2), A |
| 4C29 ED437B4C | 06040 | LD | (FBLOCK), BC |
| 4C2D ED438151 | 06050 | LD | (FENTRY), BC |
| 4C31 21F200 | 06060 | LD | HL, FBL1-FBLOCK-4 |
| 4C34 09 | 06070 | ADD | HL, BC |
| 4C35 22714D | 06080 | LD | (FBL1), HL |
| 4C38 24 | 06090 | INC | H |

```

4C39 22754E    06100      LD      (FBL2),HL
4C3C 24        06110      INC     H
4C3D 22794F    06120      LD      (FBL3),HL
4C40 24        06130      INC     H
4C41 227D50    06140      LD      (FBL4),HL
                06150      ;
4C44 21794C    06160      LD      HL,VERSCH
4C47 010A05    06170      LD      BC,FENTRY-VERSCH+2
4C4A 3A9AF5    06180      LD      A,(VF)
4C4D A7        06190      AND     A
4C4E 2813      06200      JR      Z,S3
4C50 2A9DF5    06210      LD      HL,(AE)
4C53 7C        06220      LD      A,H
4C54 B5        06230      OR      L
4C55 2003      06240      JR      NZ,S31
4C57 2ACD4C    06250      LD      HL,(ENTRY+1)
4C5A 22B151    06260 S31   LD      (FENTRY),HL
4C5D 217F51    06270      LD      HL,FENTRY-2
4C60 010400    06280      LD      BC,4
4C63 7E        06290 S3   LD      A,(HL)
4C64 CD7E52    06300      CALL   WRB
4C67 23        06310      INC     HL
4C68 0B        06320      DEC     BC
4C69 78        06330      LD      A,B
4C6A B1        06340      OR      C
4C6B 20F6      06350      JR      NZ,S3
4C6D 11A752    06360      LD      DE,FCB
4C70 CD28CE    06370      CALL   CLOSE
4C73 C28752    06380      JP      NZ,DERRGR
4C76 C3524A    06390      JP      WSTART
                06400      ;
4C79 01        06410 VERSCH DEFB    1
4C7A F4        06420      DEFB    FBL1-VERSCH-4
4C7B 0000      06430 FBLOCK DEFW    0
4C7D F3        06440      DI
4C7E 21FBC9    06450      LD      HL,0C9FBH
4C81 221240    06460      LD      (4012H),HL
4C84 213E00    06470      LD      HL,3EH
4C87 223340    06480      LD      (4033H),HL
4C8A 3EC9      06490      LD      A,0C9H
4C8C 323540    06500      LD      (4035H),A
4C8F 32E241    06510      LD      (41E2H),A
4C92 061C      06520      LD      B,1CH
4C94 215241    06530      LD      HL,4152H
4C97 36C3      06540 INIT1   LD      (HL),0C3H
4C99 23        06550      INC     HL
4C9A 363B      06560      LD      (HL),3BH
4C9C 23        06570      INC     HL
4C9D 3601      06580      LD      (HL),1
4C9F 23        06590      INC     HL
4CA0 10F5      06600      DJNZ    INIT1
4CA2 0615      06610      LD      B,15H
4CA4 36C9      06620 INIT2   LD      (HL),0C9H
4CA6 23        06630      INC     HL
4CA7 23        06640      INC     HL
4CA8 23        06650      INC     HL
4CA9 10F9      06660      DJNZ    INIT2
4CAB CD0B00    06670      CALL   OBH
4CAE 112100    06680 HIER    LD      DE,BTAB-HIER
4CB1 19        06690      ADD     HL,DE

```



```

4CB2 E5      06700      PUSH      HL
4CB3 210000   06710 VERAD2 LD      HL,0
4CB6 E3      06720 VL      EX      (SP),HL
4CB7 5E      06730      LD      E,(HL)
4CB8 23      06740      INC     HL
4CB9 56      06750      LD      D,(HL)
4CBA 23      06760      INC     HL
4CBB 4E      06770      LD      C,(HL)
4CBC 23      06780      INC     HL
4CBD 46      06790      LD      B,(HL)
4CBE 23      06800      INC     HL
4CBF 79      06810      LD      A,C
4CC0 B0      06820      OR      B
4CC1 2805    06830      JR      Z,VS2
4CC3 E3      06840      EX      (SP),HL
4CC4 EDB0    06850      LDIR
4CC6 18EE    06860      JR      VL
4CC8 E1      06870 VS2    POP     HL
4CC9 CDE241  06880      CALL    41E2H
4CCC C30000  06890 ENTRY  JP      0
4CCF        06900 BTAB    EQU     $
4D6F        06910      ORG     $+160
4D6F 01      06920 BLOCK1 DEFB    1
4D70 02      06930      DEFB    2
4D71 0000    06940 FBL1   DEFW    0
4D73        06950 BTAB1   EQU     $
4E73        06960      ORG     $+256
4E73 01      06970 BLOCK2 DEFB    1
4E74 02      06980      DEFB    2
4E75 0000    06990 FBL2   DEFW    0
4E77        07000 BTAB2   EQU     $
4F77        07010      ORG     $+256
4F77 01      07020 BLOCK3 DEFB    1
4F78 02      07030      DEFB    2
4F79 0000    07040 FBL3   DEFW    0
4F7B        07050 BTAB3   EQU     $
507B        07060      ORG     $+256
507B 01      07070 BLOCK4 DEFB    1
507C 02      07080      DEFB    2
507D 0000    07090 FBL4   DEFW    0
507F        07100 BTAB4   EQU     $
517F        07110      ORG     $+256
517F 02      07120      DEFB    2
5180 02      07130      DEFB    2
5181 0000    07140 FENTRY DEFW    0
              07150      ;
5183 CD00F4  07160 INPUT  CALL    PRINT
5186 2169F5  07170      LD      HL,TEXT16
5189 CD00F4  07180      CALL    PRINT
518C EB      07190      EX      DE,HL
518D CDD905  07200      CALL    OSD9H
5190 C3BC4A  07210      JP      CR1
              07220      ;
              07230      ; Vergleichen
              07240      ;
5193 21E8F4  07250 VERGLE LD      HL,TEXT2
5196 CD00F4  07260      CALL    PRINT
5199 CDEB51  07270      CALL    WAIT
519C 2128F5  07280      LD      HL,TEXT8
519F CD00F4  07290      CALL    PRINT

```

```

51A2 260F      07300      LD      H,READV-READY-2
51A4 2E18      07310      LD      L,18H
51A6 22C851    07320      LD      (READY),HL
51A9 ED5B85F5  07330      LD      DE,(FLEN)
51AD C36248    07340      JP      LAD
                    07350      ;
51B0 215CF5    07360 V1    LD      HL,TEXT15
51B3 FD222040  07370 V4    LD      (4020H),IY
51B7 E5        07380      PUSH    HL
51B8 CDB94A    07390      CALL    CRCR
51BB E1        07400      POP     HL
51BC CD00F4    07410      CALL    PRINT
51BF D1        07420 V2    POP     DE
51C0 C3524A    07430      JP      WSTART
                    07440      ;
                    07450      ; Fehler
                    07460      ;
51C3 2161F5    07470 V3    LD      HL,TEXT5
51C6 18EB      07480      JR      V4
                    07490      ;
                    07500      ;
                    07510      ; Unterprogramme für Lesen
                    07520      ;
51C8 1C        07530 READY INC     E
51C9 CBF8      07540      SET     7,E
51CB CDF951    07550      CALL    READB
51CE 2804      07560      JR      Z,REND
51D0 23        07570      INC     HL
51D1 77        07580      LD      (HL),A
51D2 12        07590      LD      (DE),A
51D3 D0        07600      RET     NC
51D4 C1        07610 REND  POP     BC
51D5 23        07620      INC     HL
51D6 C3FC48    07630      JP      L22
                    07640      ;
51D9 1B        07650 READV DEC     DE
51DA 7A        07660      LD      A,D
51DB B3        07670      OR      E
51DC 28D2      07680      JR      Z,V1
51DE CDF951    07690      CALL    READB
51E1 23        07700      INC     HL
51E2 28DB      07710      JR      Z,V2
51E4 38D9      07720      JR      C,V2
51E6 BE        07730      CP      (HL)
51E7 C8        07740      RET     Z
51E8 C3C351    07750      JP      V3
                    07760      ;
                    07770      ; Unterprogramme
                    07780      ;
51EB 212DF5    07790 WAIT  LD      HL,TEXT0
51EE CD00F4    07800      CALL    PRINT
51F1 CD2B00    07810 W1    CALL    INCH
51F4 FE0D      07820      CP      CR
51F6 20F9      07830      JR      NZ,W1
51F8 C9        07840      RET
                    07850      ;
51F9 D9        07860 READB EXX
51FA 0608      07870      LD      B,8
51FC 1600      07880      LD      D,0
51FE CD1652    07890 RB1  CALL    READBI

```

```

5201 3806      07900      JR      C, RB2
5203 10F9      07910      DJNZ   RB1
5205 04        07920      INC     B
5206 7A        07930 RB3    LD      A, D
5207 D9        07940      EXX
5208 C9        07950      RET
                07960      ;
5209 78        07970 RB2    LD      A, B
520A FE08      07980      CP      8
520C 28F8      07990      JR      Z, RB3
520E CB22      08000 RB4    SLA     D
5210 10FC      08010      DJNZ   RB4
5212 37        08020      SCF
5213 04        08030      INC     B
5214 18F0      08040      JR      RB3
                08050      ;
5216 C5        08060 READBI PUSH    BC
5217 DBFF      08070      IN      A, (0FFH)
5219 E601      08080      AND     01
521B 5F        08090      LD      E, A
521C 0E00      08100      LD      C, 0
521E 0D        08110 R1     DEC     C
521F 280B      08120      JR      Z, RB11
5221 DBFF      08130      IN      A, (0FFH)
5223 E601      08140      AND     01
5225 AB        08150      XOR     E
5226 1F        08160      RRA
5227 30F5      08170      JR      NC, R1
5229 C30802    08180      JP      0208H
522C 3A90F5    08190 RB11   LD      A, (BF)
522F A7        08200      AND     A
5230 2807      08210      JR      Z, RB12
5232 3A40F8    08220      LD      A, (0F840H)
5235 CB57      08230      BIT     2, A
5237 28E5      08240      JR      Z, R1
5239 C1        08250 RB12   POP     BC
523A 37        08260      SCF
523B C9        08270      RET
                08280      ;
523C 3A90F5    08290 READS  LD      A, (BF)
523F 3291F5    08300      LD      (BFS), A
5242 3E01      08310      LD      A, 1
5244 3290F5    08320      LD      (BF), A
5247 E5        08330      PUSH    HL
5248 D5        08340      PUSH    DE
5249 C5        08350      PUSH    BC
524A 3E05      08360      LD      A, 5
524C 3226F0    08370      LD      (0F026H), A
524F 3227F0    08380      LD      (0F027H), A
5252 01AA80    08390 RS1    LD      BC, 080AAH
5255 CD1652    08400 RS2    CALL    READBI
5258 381A      08410      JR      C, RS3
525A B9        08420      CP      C
525B 20F5      08430      JR      NZ, RS1
525D 3EFF      08440      LD      A, 255
525F A9        08450      XOR     C
5260 4F        08460      LD      C, A
5261 10F2      08470      DJNZ   RS2
5263 CD1652    08480 RS4    CALL    READBI
5266 380C      08490      JR      C, RS3

```

```

5268 FE66      08500      CP      66H
526A 20F7      08510      JR      NZ,PS4
526C 3E2A      08520      LD      A,'*'
526E 322644    08530      LD      (4426H),A
5271 322744    08540      LD      (4427H),A
5274 C1        08550 RS3    POP     BC
5275 D1        08560      POP     DE
5276 E1        08570      POP     HL
5277 3A91F5    08580      LD      A,(BFS)
527A 3290F5    08590      LD      (BF),A
527D C9        08600      RET
              08610      ;
CE20          08620 INIT    EQU     0CE20H
CE28          08630 CLOSE  EQU     0CE2BH
              08640 ;
527E D5        08650 WRB    PUSH    DE
527F 11A752    08660      LD      DE,FCB
5282 CD1B00    08670      CALL    1BH
5285 D1        08680      POP     DE
5286 C8        08690      RET      Z
              08700      ;
5287 0600      08710 DERROR LD      B,0
5289 04        08720 DE1    INC     B
528A D60A      08730      SUB     10
528C 30FB      08740      JR      NC,DE1
528E 05        08750      DEC     B
528F C60A      08760      ADD     A,10
5291 F5        08770      PUSH    AF
5292 78        08780      LD      A,B
5293 C630      08790      ADD     A,'0'
5295 CD28F4    08800      CALL    OUTCH
5298 F1        08810      POP     AF
5299 C630      08820      ADD     A,'0'
529B CD28F4    08830      CALL    OUTCH
529E 216DF5    08840      LD      HL,TEXT17
52A1 CD00F4    08850      CALL    PRINT
52A4 C3524A    08860      JP      WSTART
              08870      ;
              08880 ;
52A7          08890 FCB     EQU     $
              08900 ;
52C9          08910 DBUF    EQU     $+34
              08920 ;
5B78          08930 BUFFER  EQU     5B78H
              08940 ;
F400          08950      ORG     0F400H
              08960      ;
F400 7E        08970 PRINT  LD      A,(HL)
F401 A7        08980      AND     A
F402 23        08990      INC     HL
F403 2809      09000      JR      Z,PR2
F405 FEFF      09010      CP      255
F407 2812      09020      JR      Z,PR3
F409 CD28F4    09030      CALL    OUTCH
F40C 18F2      09040      JR      PRINT
              09050      ;
F40E D5        09060 PR2    PUSH    DE
F40F CD2B00    09070      CALL    INCH
F412 FE60      09080      CP      66H
F414 CC2B00    09090 PR1    CALL    Z,INCH

```

| | | | |
|-------------|--------------|------|--------------|
| F417 28FB | 09100 | JR | Z,PR1 |
| F419 D1 | 09110 | POP | DE |
| F41A C9 | 09120 | RET | |
| | 09130 | ; | |
| F41B C5 | 09140 PR3 | PUSH | BC |
| F41C 46 | 09150 | LD | B, (HL) |
| F41D 23 | 09160 | INC | HL |
| F41E 7E | 09170 PR4 | LD | A, (HL) |
| F41F CD28F4 | 09180 | CALL | OUTCH |
| F422 10FA | 09190 | DJNZ | PR4 |
| F424 23 | 09200 | INC | HL |
| F425 C1 | 09210 | POP | BC |
| F426 18D8 | 09220 | JR | PRINT |
| | 09230 | ; | |
| 002B | 09240 INCH | EQU | 002BH |
| | 09250 | ; | |
| F428 D5 | 09260 OUTCH | PUSH | DE |
| F429 F5 | 09270 | PUSH | AF |
| F42A CD3300 | 09280 | CALL | 33H |
| F42D F1 | 09290 | POP | AF |
| F42E 57 | 09300 | LD | D, A |
| F42F 3A93F5 | 09310 | LD | A, (DFLAG) |
| F432 A7 | 09320 | AND | A |
| F433 7A | 09330 | LD | A, D |
| F434 C43B00 | 09340 | CALL | NZ, 3BH |
| F437 D1 | 09350 | POP | DE |
| F438 C9 | 09360 | RET | |
| | 09370 | ; | |
| | 09380 | ; | |
| F439 7C | 09390 PRH | LD | A, H |
| F43A CD3EF4 | 09400 | CALL | PRBYTE |
| F43D 7D | 09410 | LD | A, L |
| F43E F5 | 09420 PRBYTE | PUSH | AF |
| F43F 0F | 09430 | RRCA | |
| F440 0F | 09440 | RRCA | |
| F441 0F | 09450 | RRCA | |
| F442 0F | 09460 | RRCA | |
| F443 CD47F4 | 09470 | CALL | PRNIB |
| F446 F1 | 09480 | POP | AF |
| F447 E60F | 09490 PRNIB | AND | 0FH |
| F449 FE0A | 09500 | CP | 10 |
| F44B 3802 | 09510 | JR | C, PN1 |
| F44D C607 | 09520 | ADD | A, 'A'-'9'-1 |
| F44F C630 | 09530 PN1 | ADD | A, '0' |
| F451 C328F4 | 09540 | JP | OUTCH |
| | 09550 | ; | |
| F454 CD39F4 | 09560 PRHEX | CALL | PRH |
| F457 3E48 | 09570 | LD | A, 'H' |
| F459 C328F4 | 09580 | JP | OUTCH |
| | 09590 | ; | |
| F45C 2A89F5 | 09600 PRBL | LD | HL, (LSTREC) |
| F45F 7D | 09610 | LD | A, L |
| F460 B4 | 09620 | OR | H |
| F461 C8 | 09630 | RET | Z |
| F462 D5 | 09640 | PUSH | DE |
| F463 213DF5 | 09650 | LD | HL, TEXT11 |
| F466 CD00F4 | 09660 | CALL | PRINT |
| F469 2A87F5 | 09670 | LD | HL, (LSTRAC) |
| F46C E5 | 09680 | PUSH | HL |
| F46D CD54F4 | 09690 | CALL | PRHEX |

```

F470 3E2D      09700      LD      A,'-'
F472 CD28F4    09710      CALL    OUTCH
F475 2A89F5    09720      LD      HL,(LSTREC)
F478 CD54F4    09730      CALL    PRHEX
F47B D1        09740      POP      DE
F47C A7        09750      AND      A
F47D ED52      09760      SBC     HL,DE
F47F D1        09770      POP      DE
F480 7C        09780      LD      A,H
F481 A7        09790      AND      A
F482 C0        09800      RET      NZ
F483 7D        09810      LD      A,L
F484 FE03      09820      CP      3
F486 D0        09830      RET      NC
F487 C5        09840      PUSH    BC
F488 45        09850      LD      B,L
F489 04        09860      INC     B
F48A 2181F5    09870      LD      HL,TEXT20
F48D CD00F4    09880      CALL    PRINT
F490 2A8DF5    09890      LD      HL,(LSTAD1)
F493 7E        09900 PRBL1  LD      A,(HL)
F494 CD3EF4    09910      CALL    PRBYTE
F497 3E48      09920      LD      A,'H'
F499 CD28F4    09930      CALL    OUTCH
F49C 3E20      09940      LD      A,' '
F49E CD28F4    09950      CALL    OUTCH
F4A1 23        09960      INC     HL
F4A2 10EF      09970      DJNZ    PRBL1
F4A4 C1        09980      POP      BC
F4A5 C9        09990      RET
                10000      ;
                10010      ;
                10020      ;
F4A6 EB        10030 ENDE1  EX      DE,HL
F4A7 CDACF4    10040      CALL    ENDE
F4AA EB        10050      EX      DE,HL
F4AB C9        10060      RET
                10070      ;
F4AC E5        10080 ENDE  PUSH    HL
F4AD D5        10090      PUSH    DE
F4AE 21785B    10100      LD      HL,BUFFER
F4B1 ED5885F5  10110      LD      DE,(FLEN)
F4B5 19        10120      ADD     HL,DE
F4B6 37        10130      SCF
F4B7 D1        10140      POP      DE
F4B8 ED52      10150      SBC     HL,DE
F4BA E1        10160      POP      HL
F4BB D0        10170      RET      NC
F4BC D1        10180      POP      DE
F4BD 2176F5    10190      LD      HL,TEXT19
F4C0 CD00F4    10200      CALL    PRINT
F4C3 C34E4A    10210      JP      L110
                10220      ;
                10230      ;
                10240      ; Texte
                10250      ;
F4C6 1C        10260 TEXT1 DEFB    1CH
F4C7 1F        10270      DEFB    1FH
F4C8 53        10280      DEFM    'SAVETAPE '
F4D1 41        10290 BA     DEFM    'A '

```

| | | | |
|---------|--------------|------|----------------------|
| F4D3 20 | 10300 DA | DEFM | ' ' |
| F4D5 3C | 10310 | DEFM | '<L, F, S, A, D, V>' |
| F4E7 00 | 10320 | DEFB | 0 |
| F4E8 1C | 10330 TEXT2 | DEFB | 1CH |
| F4E9 1F | 10340 | DEFB | 1FH |
| F4EA 57 | 10350 | DEFM | 'Wiedergabe' |
| F4F4 0D | 10360 | DEFB | CR |
| F4F5 00 | 10370 | DEFB | 0 |
| F4F6 0D | 10380 TEXT3 | DEFB | CR |
| F4F7 4C | 10390 | DEFM | 'Laenge' |
| F4FD 00 | 10400 | DEFB | 0 |
| F4FE 1C | 10410 TEXT4 | DEFB | 1CH |
| F4FF 1F | 10420 | DEFB | 1FH |
| F500 46 | 10430 | DEFM | 'Filename' |
| F508 00 | 10440 | DEFB | 0 |
| F509 56 | 10450 TEXT5B | DEFM | 'Verschieben' |
| F514 00 | 10460 | DEFB | 0 |
| F515 4E | 10470 TEXT5A | DEFM | 'Neue Adr' |
| F51D 00 | 10480 | DEFB | 0 |
| F51E 4C | 10490 TEXT6 | DEFM | 'L..' |
| F521 0D | 10500 TEXT6A | DEFB | CR |
| F522 00 | 10510 | DEFB | 0 |
| F523 53 | 10520 TEXT7 | DEFM | 'S..' |
| F526 0D | 10530 | DEFB | CR |
| F527 00 | 10540 | DEFB | 0 |
| F528 56 | 10550 TEXT8 | DEFM | 'V..' |
| F52B 0D | 10560 | DEFB | CR |
| F52C 00 | 10570 | DEFB | 0 |
| F52D 3C | 10580 TEXT9 | DEFM | '<ENTER>' |
| F534 0D | 10590 | DEFB | CR |
| F535 00 | 10600 | DEFB | 0 |
| F536 0D | 10610 TEXT10 | DEFB | CR |
| F537 53 | 10620 | DEFM | 'SYS: ' |
| F53C 00 | 10630 | DEFB | 0 |
| F53D 0D | 10640 TEXT11 | DEFB | CR |
| F53E 42 | 10650 | DEFM | 'Blk: ' |
| F543 00 | 10660 | DEFB | 0 |
| F544 0D | 10670 TEXT18 | DEFB | CR |
| F545 7A | 10680 | DEFM | 'zus: ' |
| F54A 00 | 10690 | DEFB | 0 |
| F54B 0D | 10700 TEXT12 | DEFB | CR |
| F54C 45 | 10710 | DEFM | 'Ety: ' |
| F551 00 | 10720 | DEFB | 0 |
| F552 3F | 10730 TEXT13 | DEFM | '?!?' |
| F555 00 | 10740 | DEFB | 0 |
| F556 42 | 10750 TEXT14 | DEFM | 'BAS: ' |
| F55B 00 | 10760 | DEFB | 0 |
| F55C 4B | 10770 TEXT15 | DEFM | 'Kein ' |
| F561 46 | 10780 TEXT5 | DEFM | 'Fehler' |
| F567 0D | 10790 | DEFB | CR |
| F568 00 | 10800 | DEFB | 0 |
| F569 20 | 10810 TEXT16 | DEFM | ' ? ' |
| F56C 00 | 10820 | DEFB | 0 |
| F56D 20 | 10830 TEXT17 | DEFM | ' Fehler' |
| F574 0D | 10840 | DEFB | CR |
| F575 00 | 10850 | DEFB | 0 |
| F576 0D | 10860 TEXT19 | DEFB | CR |
| F577 7A | 10870 | DEFM | 'zu kurz.' |
| F57F 0D | 10880 | DEFB | CR |
| F580 00 | 10890 | DEFB | 0 |

```
-----
F581 20      10900 TEXT20  DEFM    ' : '
F584 00      10910      DEFB     0
F585 0000    10920 FLEN    DEFW     0
F587 0000    10930 LSTRAC  DEFW     0
F589 0000    10940 LSTREC  DEFW     0
F58B 0000    10950 LSTAD   DEFW     0
F58D 0000    10960 LSTAD1  DEFW     0
F58F 00      10970 CSF     DEFB     0
F590 00      10980 BF      DEFB     0
F591 00      10990 BFS     DEFB     0
F592 00      11000 DF      DEFB     0
F593 00      11010 DFLAG   DEFB     0
F594 00      11020 CF      DEFB     0
F595 00      11030 FTF     DEFB     0
F596 00      11040 AFF     DEFB     0
F597 00      11050 MZ      DEFB     0
F598 0000    11060 ENDSYS  DEFW     0
F59A 00      11070 VF      DEFB     0
F59B 0000    11080 VERADR  DEFW     0
F59D 0000    11090 AE      DEFW     0
                11100      ;
4800         11110      END      START
00000 TOTAL ERRORS
19996 TEXT AREA BYTES LEFT
```


Literaturverzeichnis:
=====

- (1) Norbert Heicke, Luidger Röckrath:
Colour-Genie ROM-Listing
128 S., DIN A4, kartoniert, 1983, 59,-- DM
ISBN 3-925074-03-1
- (2) Luidger Röckrath
Maschinensprachebuch für TRS-80/Genie und Colour-Genie
149 S., DIN A5, kartoniert, 1984, 49,-- DM
ISBN 3-925074-07-4
(Inhaltsverzeichnis siehe folgende Seite)
- (3) Luidger Röckrath:
Microsoftbasic verstanden -
Konzepte, Algorithmen, Datenstrukturen (Arbeitstitel)
erscheint vor. Frühjahr 1985 im Franzis-Verlag, München
- (4) Rodney Zaks:
Die Programmierung des 280
48,-- DM
Sybex, Düsseldorf
- (5) Colour Basic-leicht gemacht
TCS GmbH, St. Augustin
150 S., DIN A4, kartoniert, 1983, 10,-- DM
ISBN 3-88965-001-7
- (6) Das technische Handbuch Colour-Genie
TCS GmbH, St. Augustin
86 S., DIN A4, kartoniert, 49,-- DM
ISBN 3-88965-002-3
- (7) Kalle Braun, Jürgen Buchmüller, Frank Seeger:
Das Colour-Genie Buch 1
TCS GmbH, St. Augustin
110 S., DIN A5, kartoniert, 1983, 39,-- DM
ISBN 3-88965-003-1
- (8) Kalle Braun, Jürgen Buchmüller, Frank Seeger:
Das Colour-Genie Buch 2
TCS GmbH, St. Augustin
118 S., DIN A5, kartoniert, 1983, 39,-- DM
ISBN 3-88965-004-X
- (9) Kalle Braun, Jürgen Buchmüller, Frank Seeger:
Das Colour-Genie Buch 3
TCS GmbH, St. Augustin
108 S., DIN A5, kartoniert, 1984, 39,-- DM
ISBN 3-88965-007-4

Alle Bücher können bei uns bezogen werden.

Inhaltsverzeichnis des Maschinensprachebuches
 =====

| | |
|---|-----|
| Teil 1: ROM-Routinen | 5 |
| 1.1 Ein-/Ausgaberoutinen | 8 |
| 1.2 Verwaltung des Programmtextes | 16 |
| 1.3 Verwaltung der Variablentabelle | 21 |
| 1.4 Arithmetik | 26 |
| 1.5 Stringverarbeitung | 34 |
| 1.6 Ausdrucksauswertung | 36 |
| 1.7 Kassettenroutinen | 38 |
| 1.8 Restart's | 45 |
| Teil 2: RAM-Adressen | 48 |
| Teil 3: Vergleich der verschiedenen ROM-Versionen | 50 |
| Teil 4: Beschreibungen der Programme | 55 |
| 4.1 Bildschirmeditor SEDIT | 56 |
| 4.2 Basicerweiterung NEWBAS | 62 |
| 4.3 Double Precision Funktionen DOUBLE | 72 |
| 4.4 Basicprogrammpacker PACK | 75 |
| 4.5 Feldaufzeichnung ARRAY | 77 |
| 4.6 Umlauterweiterung für SCRIPSIT | 80 |
| 4.7 Assembler/Disassembler 280AD | 83 |
| Teil 5: Listings der Beispielprogramme | 87 |
| 5.1 Beispiel 1: Durchschnitt eines Feldes | 87 |
| 5.2 Beispiel 2: Ausgabe einer Codetabelle | 89 |
| 5.3 Beispiel 3: POP-Anweisung | 90 |
| 5.4 Beispiel 4: Zahlenein-/ausgabe | 91 |
| 5.5 Beispiel 5: Systemfileaufzeichnung | 92 |
| Teil 6: Listings der Programme aus Teil 4 | 95 |
| 6.1 Bildschirmeditor SEDIT | 95 |
| 6.2 Basicerweiterung NEWBAS | 101 |
| 6.3 Double Precision Funktionen DOUBLE | 114 |
| 6.4 Basicprogrammpacker PACK | 124 |
| 6.5 Feldaufzeichnung ARRAY | 127 |
| 6.6 Umlauterweiterung für SCRIPSIT | 131 |
| 6.7 Assembler/Disassembler 280AD | 133 |
| 6.8 Konvertierungsprogramme /SRC <-> /MAC | 148 |

Bitte beachten Sie, daß die Teile 1, 2, 3 und 5 ohne Einschränkung für das Colour-Genie geeignet sind. Von den Programmen aus Teil 4 ist nur DOUBLE unverändert auf dem Colour-Genie lauffähig. Der Assembler/Disassembler bedarf einer Anpassung, die anderen Programme sind in einer speziellen Colour-Genie-Version im vorliegenden Buch Assemblertricks enthalten.

- Entwicklung - Vertrieb - Beratung
- Hardware - Software
- EDV - Dokumentationsverlag

Ludger Röckrath · Noppiusstraße 19 · 5100 Aachen

Neuerscheinung:

L. Röckrath:
Das Maschinensprachebuch

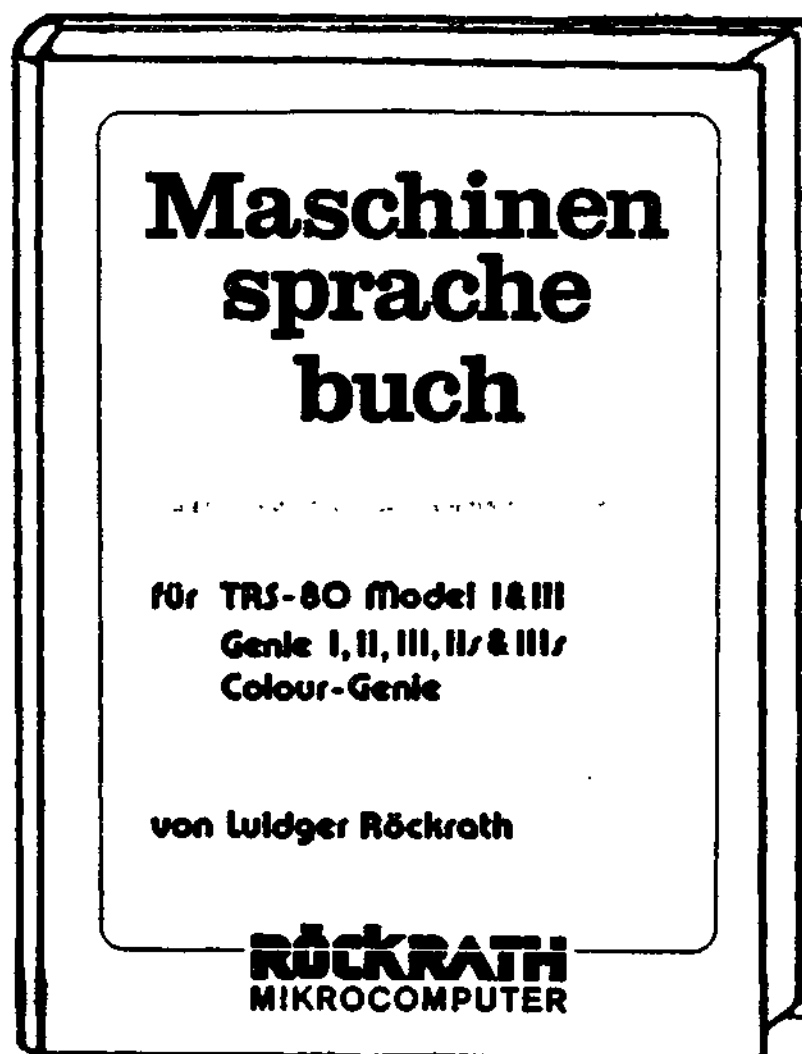
für TRS-80/Genie/Colour-Genie
(149 S., DIN-A5, kartoniert,
49 DM, ISBN 3-925074-07-4)

Das Maschinensprachebuch ist unentbehrlich für jeden, der auf einem TRS-80/Genie oder Colour-Genie Maschinenprogramme entwickelt. Es zeigt einfach und verständlich auf, wie eigene Maschinenprogramme durch Verwendung von ROM-Routinen effizient geschrieben werden können. Außerdem werden die Unterschiede zwischen den verschiedenen Systemen erläutert. Zum Schluß folgen ausführlich dokumentierte Beispielprogramme und hilfreiche Utilities im Quellprogramm.

Das Maschinenprogramm bietet unter anderem:

- ausführliche Beschreibung von ca. 70 ROM-Routinen:
 - Einsprungsadresse, Name, evtl. Unterschiede zwischen den Systemen
 - Registerbeeinflussung
 - Aufruf- und Ergebnisparameter
 - genaue und ausführliche Funktionsbeschreibung
 - 1 Beispiel für die Anwendung
- genaue Beschreibung des Aufbaus des Basicprogrammtextes, der Variablentabelle, ...
- Auflistung aller wichtigen RAM-Zellen, in denen Parameter für die ROM-Routinen übergeben werden müssen
- Vergleich der ROM's von TRS-80 Model I (alte/neue Version), TRS-80 Model III, Genie I+II, Genie III, Genie IIIs, Colour-Genie
- 5 ausführlich dokumentierte Sourcelistings von Beispielprogrammen: Durchschnitt eines Feldes, Codetabelle, POP-Anweisung, Zahlen-Ein-/Ausgabe und System-File-Aufzeichnung auf Kassette.
- Sourcelistings und Beschreibungen zahlreicher Utilities, die programmiertechniken aufzeigen und sinnvoll eingesetzt werden können: komfortabler Bildschirmeditor SEDIT / Basicerweiterung NEWBAS / Erweiterung aller Funktionen auf doppelte Genauigkeit DOUBLE / Basicprogrammpacker PACK / Feldaufzeichnung ARRAY / Umlaut- und Druckersteuerungserweiterung für SCRIPSIT / Assembler/Disassembler/Editor/Monitor Z80AD (nicht alle Programme auf TRS-80-Model-III/Colour-Genie lauffähig.)
- Erläuterung aller Methoden des Betriebssystemeingriffes mit Beispielen: Eingriff über RST-Vektoren, Disk-Basic-Vektoren, Treibervektoren im DCB, Befehlserweiterung durch Eingriff in die Interpretationsschleife.

RÖCKRATH MIKROCOMPUTER



Luidger Röckrath:

Assemblertricks auf dem Colour-Genie

=====

Dieses Buch zeigt alle Tricks auf, mit denen auf dem Colour-Genie effektiv in Assembler programmiert werden kann.

Damit wird der Zugriff vom Assemblerprogramm auf freiprogrammierbare Zeichen, hochauflösende Grafik, PSG, CRTC und Diskette spielend einfach. Durch eine genaue Beschreibung von vielen ROM-Routinen können schnell Maschinenprogramme hoher Leistungsfähigkeit geschrieben werden. Zahlreiche Beispiele (darunter eine Erweiterung des Bildformates auf 30 Zeilen a 50 Zeichen und ein Programm zum Lesen von disk-relativ adressierten Sektoren) zeigen Programmiertechniken und sinnvolle Anwendung der vorgestellten Routinen im Zusammenhang auf.

Schließlich folgen die Beschreibungen und Quellprogramme zahlreicher professioneller Utilities, vom Bildschirmeditor SEDIT bis zu den beiden universellen Kopierprogrammen COPBIT und SAVETAPE.

ISBN 3-925074-08-2